Spatial Wireless Channel Prediction under Location Uncertainty

L. Srikar Muppirisetty, Tommy Svensson, *Senior Member, IEEE*, and Henk Wymeersch, *Member, IEEE*

***Abstract*—Spatial wireless channel prediction is important for future wireless networks, and in particular for proactive resource allocation at different layers of the protocol stack. Various sources of uncertainty must be accounted for during modeling and to provide robust predictions. We investigate two channel prediction frameworks, classical Gaussian processes (cGP) and uncertain Gaussian processes (uGP), and analyze the impact of location uncertainty during learning/training and prediction/testing, for scenarios where measurements uncertainty are dominated by large-scale fading. We observe that cGP generally fails both in terms of learning the channel parameters and in predicting the channel in the presence of location uncertainties. In contrast, uGP explicitly considers the location uncertainty. Using simulated data, we show that uGP is able to learn and predict the wireless channel.**

***Index Terms*—Gaussian processes, uncertain inputs, location uncertainty, spatial predictability of wireless channels.**

I. Introduction

LOCATION-based resource allocation schemes are expected to become an essential element of emerging 5G networks, as 5G devices will have the capability to accurately self-localize and predict relevant channel quality metrics (CQM) [1]–[3] based on crowd-sourced databases. The geo-tagged CQM (including, e.g., received signal strength, delay spread, and interference levels) from users enables the construction of a dynamic database, which in turn allows the prediction of CQM at arbitrary locations and future times. Current standards are already moving in this direction through the so-called minimization of drive test (MDT) feature in 3GPPP Release 10 [4]. In MDT, users collect radio measurements and associated location information in order to assess network performance. In terms of applications, prediction of spatial wireless channels (e.g., through radio environment maps) and its utilization in resource allocation can reduce overheads and delays due to the ability to predict channel quality beyond traditional time scales [2]. Exploitation of location-aware CQM is relevant for interference management in two-tier cellular networks [5], coverage hole detection and prediction [6], cooperative spectrum sensing in cognitive radios [7], anticipatory networks for predictive resource allocation [3], and proactive caching [8].

In order to predict location-dependent radio propagation channels, we rely on mathematical models, in which the physical environment, including the locations of transmitter and receiver, play an important role. The received signal power in a wireless channel is mainly affected by three major dynamics, which occur at different length scales: path-loss, shadowing, and small-scale fading [9]. Small-scale fading decorrelates within tens of centimeters (depending on the carrier frequency), making it infeasible to predict based on location information. On the other hand, shadowing is correlated up to tens of meters, depending on the propagation environment (e.g., 50–100 m for outdoor [9] and 1–2 m for indoor environments [10]). Finally, path-loss, which captures the deterministic decay of power with distance, is a deterministic function of the distance to the transmitter. In rich scattering environments, the measurements average small-scale fading either in frequency or space provided sufficient bandwidth or number of antennas [10]. Thus, provided that measurements are dominated by large-scale fading, location-dependent models for path-loss and shadowing can be developed based on the physical properties of the wireless channel. With the help of spatial regression tools, these large-scale channel components can be predicted at other locations and used for resource allocation [1]. However, since localization is subject to various error sources (e.g., the global positioning system (GPS) gives an accuracy of around 10 m [11] in outdoor scenarios, while ultra-wide band (UWB) systems can give submeter accuracy), there is a fundamental need to account for location uncertainties when developing spatial regression tools.

Spatial regression tools generally comprise a training/learning phase, in which the underlying channel parameters are estimated based on the available training database, and a testing/prediction phase, in which predictions are made at test locations, given learned parameters and the training database. Among such tools, Gaussian processes (GP) is a powerful and commonly used regression framework, since it is generally considered to be the most flexible and provides prediction uncertainty information [12]. Two important limitations of GP are its computational complexity [13]–[16] and its sensitivity to uncertain inputs [14], [17]–[21]. To alleviate the computational complexity, various sparse GP techniques have been proposed in [13]–[15], while online and distributed GP were treated in [16], [22], [23] and [24]–[26], respectively. The impact of input uncertainty was studied in [17], [18], which showed that GP was adversely affected, both in training and testing, by input uncertainties. The input uncertainty in our case corresponds to location uncertainty.

No framework has yet been developed to mathematically characterize and understand the spatial predictability of wireless channels with location uncertainty. In this paper, we build

The authors are with the Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden, e-mail: {srikar.muppirisetty, henkw, tommy.svensson}@chalmers.se.

This research was supported, in part, by the European Research Council, under Grant No. 258418 (COOPNET), and the Swedish Research Council VR under the project Grant No. 621-2009-4555 (Dynamic Multipoint Wireless Transmission).

on and adapt the framework from [17], [18] to CQM prediction in wireless networks. Our main contributions are as follows:

- We show that not considering location uncertainty leads to poor learning of the channel parameters and poor prediction of CQM values at other locations, especially when location uncertainties are heterogeneous;
- We relate and unify existing GP methods that account for uncertainty during both learning and prediction, by operating directly on an input set of distributions, rather than an input set of locations;
- We describe and delimit proper choices for mean functions and covariance functions in this unified framework, so as to incorporate location uncertainty in both learning and prediction; and
- We demonstrate the use of the proposed framework for simulated data and apply it to a spatial resource allocation application.

The remainder of the paper is structured as follows. Section III presents the channel model and details the problem description for location-dependent channel prediction with location uncertainty. In Section IV, we review channel learning and prediction in the classical GP (cGP) setup with no localization errors. Section V details learning and prediction procedures using the proposed GP framework that accounts for uncertainty on training and test locations, termed uncertain GP (uGP). Finally, numerical results are given in Section VI in addition to a resource allocation example, followed by our conclusions in Section VII.

Notation: Vectors and matrices are written in bold (e.g., a vector $\mathbf{k}$ and a matrix $\mathbf{K}$); $\mathbf{K}^{\mathrm{T}}$ denotes transpose of $\mathbf{K}$; $|\mathbf{K}|$ denotes determinant of $\mathbf{K}$; $[\mathbf{K}]_{ij}$ denotes entry (i,j) of $\mathbf{K}$; $\mathbf{I}$ denotes identity matrix of appropriate size; $\mathbf{1}$ and $\mathbf{0}$ are vectors of ones and zeros, respectively, of appropriate size; $\|.\|$ denotes L_2-norm unless otherwise stated; $\mathbb{E}[.]$ denotes the expectation operator; $\mathrm{Cov}[.]$ denotes covariance operator (i.e., $\mathrm{Cov}[\mathbf{y}_1,\mathbf{y}_2] = \mathbb{E}[\mathbf{y}_1\mathbf{y}_2^{\mathrm{T}}] - \mathbb{E}[\mathbf{y}_1]\,\mathbb{E}[\mathbf{y}_2]^{\mathrm{T}}$); $\mathcal{N}(\mathbf{x};\mathbf{m},\mathbf{\Sigma})$ denotes a Gaussian distribution evaluated in $\mathbf{x}$ with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$ and $\mathbf{x} \sim \mathcal{N}(\mathbf{m},\mathbf{\Sigma})$ denotes that $\mathbf{x}$ is drawn from a Gaussian distribution with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$. Important symbols used in the paper are: $\mathbf{x}_i \in \mathbb{R}^2$ is an exact, true location; $\mathbf{u}_i \in \mathbb{R}^D$, $D > 2$ is a vector that describes (e.g., in the form of moments) the location distribution $p(\mathbf{x}_i)$. For example in the case of Gaussian distributed localization error, $p(\mathbf{x}) = \mathcal{N}(\mathbf{x};\mathbf{z},\mathbf{\Sigma})$, then a possible choice is $\mathbf{u} = [\mathbf{z}^{\mathrm{T}}, \mathrm{vec}[\mathbf{\Sigma}]]^{\mathrm{T}}$, where $\mathrm{vec}[\mathbf{\Sigma}]$ stacks all the elements of $\mathbf{\Sigma}$ in a vector. Finally, $\mathbf{z}_i = \phi(\mathbf{u}_i) \in \mathbb{R}^2$ is a location estimate extracted from $\mathbf{u}_i$ through a function $\phi(\cdot)$ (e.g., the mean or mode).

II. Related Work

First, we give an overview of the literature on GP with uncertain inputs. One way to deal with the input noise is through linearizing the output around the mean of the input [19], [21]. In [21], the input noise was viewed as extra output noise by linearization at each point and this is proportional to the squared gradient of the GP posterior mean. However, the proposed method works under the condition of constant-variance input noise. In [19], a Delta method was used for linearization under the assumption of Gaussian distributed inputs and proposed a corrected covariance function that accounts for the input noise variance. For Gaussian distributed test inputs and known training inputs, the exact and approximate moments of the GP posterior was examined for various forms of covariance functions [18]. Training on Gaussian distributed input points by calculating the expected covariance matrix was studied in [17], [18]. Two approximations were evaluated in [27], first a joint maximization of joint posterior on uncertain inputs and hyperparameters (leading to over-fitting), and second using a stochastic expectation–maximization algorithm (at a high computational cost).

We now review previous works on GP for channel prediction, which include spatial correlation of shadowing in cellular [28] and ad-hoc networks [29], as well as tracking of transmit powers of primary users in a cognitive network [23]. In [28], GP was shown to model spatially correlated shadowing to predict shadowing and path-loss at any arbitrary location. A multi-hop network scenario was considered [29], and shadowing was modeled using a spatial loss field, integrated along a line between transmitter and receiver. In [23], a cognitive network setting was evaluated, in which the transmit powers of the primary users were tracked with cooperation among the secondary users. For this purpose a distributed radio channel tracking framework using Kriged Kalman filter was developed with location information. A study on the impact of underlying channel parameters on the spatial channel prediction variance using GP was presented in [30]. A common assumption in [23], [28]–[30] was the presence of perfect location information. This assumption was partially removed in [31], which extends [30] to include the effect of localization errors on spatial channel prediction. It was found that channel prediction performance was degraded when location errors were present, in particular when either the shadowing standard deviation or the shadowing correlation were large. However, [31] did not tackle combined learning and prediction under location uncertainty. The only work that explicitly accounts for location uncertainty was [20], in which the Laplace approximation was used to obtain a closed-form analytical solution for the posterior predictive distribution. However, [20] did not consider learning of parameters in presence of location uncertainty.

III. System Model

A. Channel Model

Consider a geographical region $\mathcal{A} \subseteq \mathbb{R}^2$, where a source node is located at the origin and transmits a signal with power P_{TX} to a receiver located at $\mathbf{x}_i \in \mathcal{A}$ through a wireless propagation channel. The received radio signal is affected mainly by distance-dependent path-loss, shadowing due to obstacles in the propagation medium, and small-scale fading due to multipath effects. The received power $P_{\mathrm{RX}}(\mathbf{x}_i)$ can be expressed as [32, Chap. 2]

$$P_{\mathrm{RX}}(\mathbf{x}_i) = P_{\mathrm{TX}}\, g_0\, \|\mathbf{x}_i\|^{-\eta}\, \psi(\mathbf{x}_i)\, |h(\mathbf{x}_i)|^2, \qquad (1)$$

where g_0 is a constant that captures antenna and other propagation gains, η is the path-loss exponent, $\psi(\mathbf{x}_i)$ is the location-dependent shadowing and $h(\mathbf{x}_i)$ is the small-scale fading. We assume measurements average[1] small-scale fading, either in time (measurements taken over a time window), frequency (measurements represent average power over a large frequency band), or space (measurements taken over multiple antennas) [10], [33]. Therefore, the resulting received signal power from the source node to a receiver node i can be expressed in dB scale as

$$P_{\mathrm{RX}}(\mathbf{x}_i)[\mathrm{dBm}] = L_0 - 10\,\eta\,\log_{10}(\|\mathbf{x}_i\|) + \Psi(\mathbf{x}_i), \qquad (2)$$

where $L_0 = P_{\mathrm{TX}}[\mathrm{dBm}] + G_0$ with $G_0 = 10\,\log_{10}(g_0)$ and $\Psi(\mathbf{x}_i) = 10\,\log_{10}(\psi(\mathbf{x}_i))$. A common choice for modeling shadowing in wireless systems is through a log-normal distribution, i.e., $\Psi(\mathbf{x}_i) \sim \mathcal{N}(0, \sigma_\Psi^2)$, where σ_Ψ^2 is the shadowing variance. Shadowing $\Psi(\mathbf{x}_i)$ is spatially correlated, with well-established correlation models [34], among which the Gudmundson model is widely used [35]. Let y_i be the scalar[2] observation of the received power at node i, which is written as $y_i = P_{\mathrm{RX}}(\mathbf{x}_i) + n_i$, where n_i is a zero mean additive white Gaussian noise with variance σ_n^2. For the sake of notational simplicity, we do not consider a three-dimensional layout, the impact of non-uniform antenna gain patterns, or distance-dependent path-loss exponents.

B. Location Error Model

In practice, nodes may not have access to their true location $\mathbf{x}_i$, but only to a distribution $p(\mathbf{x}_i)$[3]. The distribution $p(\mathbf{x}_i)$ is obtained from the positioning algorithm in the devices, and depends on the specific positioning technology (e.g., for GPS the distribution $p(\mathbf{x}_i)$ can be modeled as a Gaussian). We will assume that all distributions $p(\mathbf{x}_i)$ come from a given family of distributions (e.g., all bivariate Gaussian distributions). These distributions can be described by a finite set of parameters, $\mathbf{u}_i \in \mathbb{R}^D$, $D > 2$, e.g., a mean and a covariance matrix for Gaussian distributions. The set of descriptions of all distributions from the given family is denoted by $\mathcal{U} \subset \mathbb{R}^D$. Within this set, the set of all delta Dirac distributions over locations is denoted by $\mathcal{X} \subset \mathcal{U}$. Note that $\mathcal{X}$ is equivalent to the set $\mathcal{A}$ of possible locations. Finally, we introduce a function $\phi : \mathcal{U} \to \mathcal{A}$ that extracts a position estimate from the distribution (in our case chosen as the mean), and denote $\mathbf{z}_i = \phi(\mathbf{u}_i) \in \mathcal{A}$. We will generally make no distinction between a distribution $p(\mathbf{x}_i)$ and its representation $\mathbf{u}_i$.

C. Problem Statement

We assume a central coordinator, which collects a set of received power measurements $\mathbf{y} = [y_1, \ldots, y_N]^T$ with respect to a common source from N nodes, along with their corresponding location distributions $\mathbf{U} = [\mathbf{u}_1^{\mathrm{T}}, \mathbf{u}_2^{\mathrm{T}}, \ldots, \mathbf{u}_N^{\mathrm{T}}]^{\mathrm{T}}$. Our goals are to perform

[1]If measurements cannot average over small-scale fading, the proposed framework from this paper cannot be applied.

[2]Vector measurements are also possible (e.g., from multiple base stations), but not considered here for the sake of clarity.

[3]$p(\mathbf{x}_i)$ is used for $p(\mathbf{x} = \mathbf{x}_i)$ for notational simplicity.

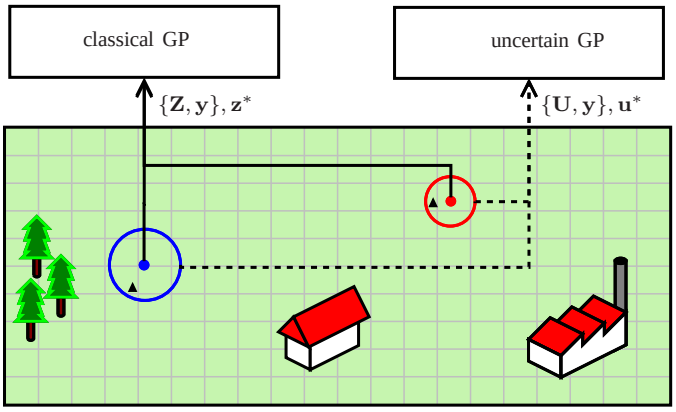

Figure 1. High-level comparison between cGP and uGP. The inputs to cGP during learning are observations $\mathbf{Y}$ and estimates $\mathbf{Z}$ of the (unobserved) actual locations $\mathbf{X}$ where those observations have been taken. $\mathbf{Z}$ is obtained through a positioning system. The true locations $\mathbf{X}$ are marked with a triangle and are generally different from the estimated locations $\mathbf{Z}$, marked with a blue and red dot. During prediction, cGP predicts received power at an estimated test location, $\mathbf{z}^*$. In contrast, uGP considers the distribution of the locations $\mathbf{X}$, described by $\mathbf{U}$ (and depicted by the red and blue circle), during learning. During prediction, uGP utilizes the distribution $\mathbf{u}^*$ of the test location. Note that the amount of uncertainty (radius of the circle) can change.

1) *Learning*: construct a spatial model (through estimating model parameters $\boldsymbol{\theta}$, to be defined later) of the received power based on the measurements;
2) *Prediction*: determine the predictive distribution $p(P_{\mathrm{RX}}(\mathbf{x}_*)|\mathbf{y}, \mathbf{U}, \hat{\boldsymbol{\theta}}, \mathbf{x}_*)$ of the power in test locations $\mathbf{x}_*$ and the distribution of the expected[4] received power, $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{y}, \mathbf{U}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*)$, for test location distributions $\mathbf{u}_*$.

We will consider two methods for learning and prediction: classical GP (Section IV), which ignores location uncertainty and only considers $\mathbf{z}_i = \phi(\mathbf{u}_i)$, and uncertain GP (Section V), which is a method that explicitly accounts for location uncertainty. We introduce $\mathbf{X} = [\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, \ldots, \mathbf{x}_N^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{Z} = [\mathbf{z}_1^{\mathrm{T}}, \mathbf{z}_2^{\mathrm{T}}, \ldots, \mathbf{z}_N^{\mathrm{T}}]^{\mathrm{T}}$ as the collection of true and estimated locations respectively. A high level comparison of cGP and uGP is shown in Fig. 1, where cGP operates on $\mathbf{Z}$ and $\mathbf{Y}$, while uGP operates on $\mathbf{U}$ and $\mathbf{Y}$.

IV. Channel Prediction with Classical GP

We first present cGP under the assumption that all locations during learning and prediction are known exactly, based on [12], [36]. Later in this section, we will discuss the impact of location uncertainties on cGP in learning/training and prediction/testing.

A. cGP without Location Uncertainty

We designate $\mathbf{x}_i \in \mathcal{A}$ as the *input* variable, and $P_{\mathrm{RX}}(\mathbf{x}_i)$ as the *output* variable. We model $P_{\mathrm{RX}}(\mathbf{x}_i)$ as a GP with mean function $\mu(\mathbf{x}_i) : \mathcal{A} \to \mathbb{R}$ and a positive semidefinite covariance function $C(\mathbf{x}_i, \mathbf{x}_j) : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$, and we write

$$P_{\mathrm{RX}}(\mathbf{x}_i) \sim \mathcal{GP}(\mu(\mathbf{x}_i), C(\mathbf{x}_i, \mathbf{x}_j)), \qquad (3)$$

[4]Here, $P_{\mathrm{RX}}(\mathbf{u}_*)$ should be interpreted as the expected received power, $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{y}, \mathbf{U}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*) = \int p(P_{\mathrm{RX}}(\mathbf{x}_*)|\mathbf{y}, \mathbf{U}, \hat{\boldsymbol{\theta}}, \mathbf{x}_*) p(\mathbf{x}_*) \mathrm{d}\mathbf{x}_*$, where $p(\mathbf{x}_*)$ is described by $\mathbf{u}_*$

where $\mathcal{GP}$ stands for a Gaussian process. The mean function[5] is defined as $\mu(\mathbf{x}_i) = \mathbb{E}_{\Psi(\mathbf{x}_i)}[P_{\mathrm{RX}}(\mathbf{x}_i)] = L_0 - 10\,\eta\,\log_{10}(\|\mathbf{x}_i\|)$, due to (2). The covariance function is defined as $C(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Cov}[P_{\mathrm{RX}}(\mathbf{x}_i), P_{\mathrm{RX}}(\mathbf{x}_j)]$. We will consider a class of covariance functions of the form:

$$C(\mathbf{x}_i, \mathbf{x}_j) = \sigma_\Psi^2 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^p}{d_c^p}\right) + \delta_{ij}\,\sigma_{\mathrm{proc}}^2, \quad (4)$$

where $\delta_{ij} = 1$ for $i = j$ and zero otherwise, $p \geq 1$, d_c is the correlation distance of the shadowing, and σ_{proc} captures any noise variance term that is not due to measurement noise (more on this later). Setting $p = 1$ in (4), gives the exponential covariance function that is commonly used to describe the covariance properties of shadowing [35], and $p = 2$, gives the squared exponential covariance function that will turn out to be useful in Section V.C. Note that the mean and covariance depend on

$$\boldsymbol{\theta} = [\sigma_n, \sigma_{\mathrm{proc}}, d_c, L_0, \eta, \sigma_\Psi], \quad (5)$$

which may not be known a priori.

1) Learning: The objective during learning is to infer the model parameters $\boldsymbol{\theta}$ from observations $\mathbf{y}$ of the received power at N *known* locations $\mathbf{X}$. The resulting training database is thus $\{\mathbf{X}, \mathbf{y}\}$. Due to the GP model, the joint distribution of the N training observations exhibits a Gaussian distribution

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}(\mathbf{X}), \mathbf{K}), \quad (6)$$

where $\boldsymbol{\mu}(\mathbf{X}) = [\mu(\mathbf{x}_1), \mu(\mathbf{x}_2), \ldots, \mu(\mathbf{x}_N)]^{\mathrm{T}}$ is the mean vector and $\mathbf{K}$ is the covariance matrix of the measured received powers, with entries $[\mathbf{K}]_{ij} = C(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2\,\delta_{ij}$. The model parameters can be learned through maximum likelihood estimation, given the training database $\{\mathbf{X}, \mathbf{y}\}$, by minimizing the negative log-likelihood function with respect to $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{-\log(p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}))\}. \quad (7)$$

The negative log-likelihood function is usually not convex and may contain multiple local optima. Additional details on the learning process are provided later. Once $\hat{\boldsymbol{\theta}}$ is determined from $\{\mathbf{X}, \mathbf{y}\}$, the training process is complete.

2) Prediction: After learning, we can determine the predictive distribution of $P_{\mathrm{RX}}(\mathbf{x}_*)$ at a new and arbitrary test location $\mathbf{x}_*$, given the training database $\{\mathbf{X}, \mathbf{y}\}$ and $\hat{\boldsymbol{\theta}}$. We first form the joint distribution

$$\begin{bmatrix} \mathbf{y} \\ P_{\mathrm{RX}}(\mathbf{x}_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{X}) \\ \mu(\mathbf{x}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_*^{\mathrm{T}} & k_{**} \end{bmatrix}\right), \quad (8)$$

where $\mathbf{k}_*$ is the $N \times 1$ vector of cross-covariances $C(\mathbf{x}_*, \mathbf{x}_i)$ between the received power at $\mathbf{x}_*$ and at the training locations $\mathbf{x}_i$, and $k_{**} = C(\mathbf{x}_*, \mathbf{x}_*)$ is the prior variance (i.e., the variance in the absence of measurements), given by $C(\mathbf{x}_*, \mathbf{x}_*)$. Conditioning on the observations $\mathbf{y}$, we obtain the Gaussian posterior distribution $p(P_{\mathrm{RX}}(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{x}_*)$ for the test location $\mathbf{x}_*$. The mean ($\bar{P}_{\mathrm{RX}}(\mathbf{x}_*)$) and variance ($V_{\mathrm{RX}}(\mathbf{x}_*)$) of this distribution turn out to be [12]

$$\begin{aligned} \bar{P}_{\mathrm{RX}}(\mathbf{x}_*) &= \mu(\mathbf{x}_*) + \mathbf{k}_*^{\mathrm{T}}\,\mathbf{K}^{-1}\,(\mathbf{y} - \boldsymbol{\mu}(\mathbf{X})) \quad (9) \\ &= \mu(\mathbf{x}_*) + \sum_{i,j=1}^{N} [\mathbf{K}^{-1}]_{ij}\,(y_j - \mu(\mathbf{x}_j))\,C(\mathbf{x}_*, \mathbf{x}_i) \\ &= \mu(\mathbf{x}_*) + \sum_{i=1}^{N} \beta_i\,C(\mathbf{x}_*, \mathbf{x}_i). \\ V_{\mathrm{RX}}(\mathbf{x}_*) &= k_{**} - \mathbf{k}_*^{\mathrm{T}}\,\mathbf{K}^{-1}\,\mathbf{k}_* \quad (10) \\ &= k_{**} - \sum_{i,j=1}^{N} [\mathbf{K}^{-1}]_{ij}\,C(\mathbf{x}_*, \mathbf{x}_i)\,C(\mathbf{x}_*, \mathbf{x}_j), \end{aligned}$$

where $\beta_i = \sum_{j=1}^{N} [\mathbf{K}^{-1}]_{ij}(y_j - \mu(\mathbf{x}_j))$. In (9), $\mu(\mathbf{x}_*)$ corresponds to the deterministic path-loss component at $\mathbf{x}_*$, which is corrected by a term involving the database and the correlation between the measurements at the training locations and the test location. In (10), we see that the prior variance k_{**} is reduced by a term that accounts for the correlation of nearby measurements.

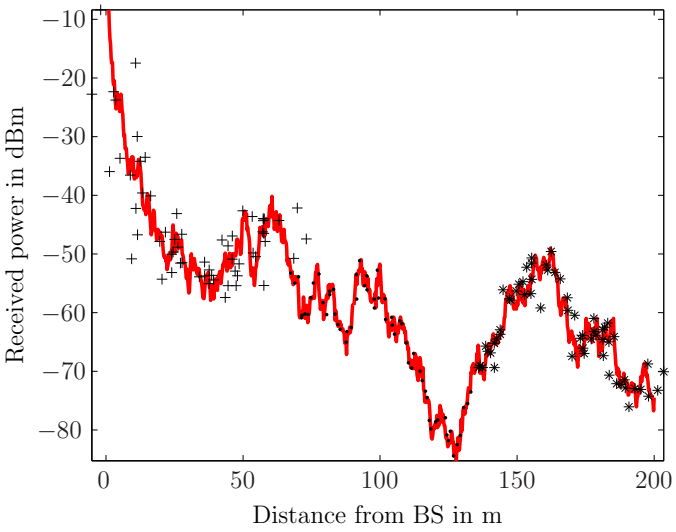

Figure 2. Impact of location uncertainty for a one-dimensional example: the red curve depicts the received signal power $P_{\mathrm{RX}}(\mathbf{x})$ as a function of $\mathbf{x}$ (or equivalently, the distance to the base station), while the markers show $P_{\mathrm{RX}}(\mathbf{x}_i)$ as a function of $\mathbf{z}_i = \phi(\mathbf{u}_i)$. Training measurements are grouped into three regions: (+) corresponds to high uncertainty, (·) corresponds to low uncertainty, and (*) corresponds to medium uncertainty, respectively. The location uncertainty results in output noise.

B. cGP with Location Uncertainty

Now let us consider the case when the nodes do not have access to their true location $\mathbf{x}_i$, but only to a distribution $p(\mathbf{x}_i)$, which is described by $\mathbf{u}_i \in \mathcal{U}$. Fig. 2 illustrates the impact of location uncertainties assuming Gaussian location errors for a one-dimensional example. The figure shows (in red) the true received power $P_{\mathrm{RX}}(\mathbf{x})$ as a function of $\mathbf{x}$ as well as the measured power $P_{\mathrm{RX}}(\mathbf{x}_i)$ as a function of $\mathbf{z}_i = \phi(\mathbf{u}_i)$ for a discrete number of values of $\mathbf{u}$, shown as markers. To clearly illustrate the impact of different amounts on uncertainty on the position, we have artificially created three regions: high location uncertainty close to the transmitter, medium

[5]Other ways of including the mean function in the model are possible, such as to include it in the covariance structure, and transform the prior model to a zero-mean GP prior [12].

location uncertainty far away, and low location uncertainty for intermediate distances. When there is no location uncertainty (70 m until 140 m from the transmitter), $\mathbf{z}_i \approx \mathbf{x}_i$, so $P_{\mathrm{RX}}(\mathbf{z}_i) \approx P_{\mathrm{RX}}(\mathbf{x}_i)$, and hence the black dots coincide with the red curve. For medium and high uncertainty, $\mathbf{z}_i$ can differ significantly from $\mathbf{x}_i$, so the data point with coordinates $[\mathbf{z}_i, P_{\mathrm{RX}}(\mathbf{x}_i)]$ can lie far away from the red curve, especially for high location uncertainty (distances below 70 m). From Fig. 2 it is clear that the input uncertainty manifests itself as output noise, with a variance that grows with increasing location uncertainty[6]. This output noise must be accounted for in the model during learning and prediction. When these uncertainties are ignored, both learning and prediction will be of poor quality, as described below.

1) Learning from uncertain training locations: In this case, the training database $\{\mathbf{Z}, \mathbf{y}\}$ comprises locations $\mathbf{z}_i = \phi(\mathbf{u}_i)$ and power measurements $y_i = P_{\mathrm{RX}}(\mathbf{x}_i) + n_i$ at the true (but unknown) locations $\mathbf{x}_i$. The measurements will be of the form shown in Fig. 2. The estimated model parameters $\hat{\boldsymbol{\theta}}$ can take two forms: (i) assign very short correlation distances $\hat{d}_c$, large $\hat{\sigma}_\Psi$, and small $\hat{\sigma}_{\mathrm{proc}}$, as some seemingly nearby events will appear uncorrelated: or (ii) assign larger correlation distances $\hat{d}_c$, smaller $\hat{\sigma}_\Psi$, and explain the measurements by assigning a higher value to $\hat{\sigma}_{\mathrm{proc}}$ [21]. In the first case, correlations between measurement cannot be exploited, so that during prediction, the posterior mean will be close to the prior mean and the posterior variance will be close to the prior variance. In the second case, predictions will be better, as correlations can be exploited to reduce the posterior variance. However, the model must explain different levels of input uncertainty with a single covariance function, which can make no distinctions between locations with low, medium, or high uncertainty. This will lead to poor performance when location error statistics differ from node to node.

2) Prediction at an uncertain test location: In the case where training locations are exactly known (i.e., $\mathbf{z}_i = \mathbf{x}_i$, $\forall i$), we may want to predict the power at an uncertain test location $\mathbf{u}_*$, made available to cGP in the form $\mathbf{z}_* = \phi(\mathbf{u}_*)$, while the true test location $\mathbf{x}_*$ is not known. This scenario can occur when a mobile user relies on a low-quality localization system and reports an erroneous location estimate to the base station. The wrong location has impact on the predicted posterior distribution since the predicted mean $\mu(\mathbf{z}_*)$ will differ from the correct mean $\mu(\mathbf{x}_*)$. In addition, $\mathbf{k}_*$ will contain erroneous entries: the j-th entry will be too small when $\|\mathbf{z}_* - \mathbf{x}_j\| > \|\mathbf{x}_* - \mathbf{x}_j\|$ and too large when $\|\mathbf{z}_* - \mathbf{x}_j\| < \|\mathbf{x}_* - \mathbf{x}_j\|$. This will affect both the posterior mean (9) and variance (10). In the case were training locations are also unknown, i.e., $\mathbf{Z} \neq \mathbf{X}$, and $\mathbf{z}_* \neq \mathbf{x}_*$, these effects are further exacerbated by the improper learning of $\boldsymbol{\theta}$.

V. Channel Prediction with Uncertain GP

In the previous section, we have argued that cGP is unable to learn and predict properly when training or test locations are not known exactly, especially when location error statistics are heterogeneous. In this section, we explore several possibilities to explicitly incorporate location uncertainty. We recall that $\mathcal{U}$ denotes the set of all distributions over the locations in the environment $\mathcal{A}$, while $\mathcal{X} \subset \mathcal{U}$ represents the delta Dirac distributions over the positions and has a one-to-one mapping to $\mathcal{A}$.

We will describe three approaches. First, a Bayesian approach where the uncertain input (i.e., the uncertain location) is marginalized, leading to a non-Gaussian output (i.e., the received power) distribution. Second, we derive a Gaussian approximation of the output distribution through moment matching and detail the corresponding learning and prediction expressions. From these expressions, the concepts of expected mean function and expected covariance function naturally appear. Finally, we discuss uncertain GP, which is a Gaussian process with input $\mathbf{u}$ from input set $\mathcal{U}$ and output y. We will relate these three approaches in a unified view. For each approach, we detail the quality of the solution and the computational complexity. We note that other approaches exist, e.g., through linearizing the output around the mean of the input [19], [21], but they are limited to mildly non-linear scenarios.

A. Bayesian Approach

In a Bayesian context, we learn and predict by integrating the respective distributions over the uncertainty of the training and test locations. As this method will involve Monte Carlo integration, we will refer to it as Monte Carlo GP (MCGP).

1) Learning: Given the training database $\{\mathbf{U}, \mathbf{y}\}$, the likelihood function with uncertain training locations $p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta})$ is obtained by integrating[7] $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ over the random training locations:

$$p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\, p(\mathbf{X})\, \mathrm{d}\mathbf{X}, \tag{11}$$

where $p(\mathbf{X}) = \prod_{i=1}^{N} p(\mathbf{x}_i)$. As there is generally no closed-form expression for the integral (11), we resort to a Monte Carlo approach by drawing M i.i.d. samples $\mathbf{X}^{(m)} \sim p(\mathbf{X})$, $1 \leq m \leq M$ so that

$$\begin{aligned} p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta}) &\approx \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{y}|\mathbf{X}^{(m)}, \boldsymbol{\theta}) \\ &= \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}(\mathbf{X}^{(m)}), \mathbf{K}^{(m)}), \end{aligned} \tag{12}$$

where $[\mathbf{K}^{(m)}]_{ij} = C(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)}) + \sigma_n^2\, \delta_{ij}$ and $\boldsymbol{\mu}(\mathbf{X}^{(m)}) = [\mu(\mathbf{x}_1^{(m)}), \mu(\mathbf{x}_2^{(m)}), \ldots, \mu(\mathbf{x}_N^{(m)})]^{\mathrm{T}}$. Finally, an estimate of $\boldsymbol{\theta}$ can be found by minimizing the negative log-likelihood function

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{-\log(p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta}))\}, \tag{13}$$

which has to be solved numerically.

[6] In fact, the output noise induced by location uncertainty will also depend on the slope of $P_{\mathrm{RX}}(\mathbf{x}_i)$ around $\mathbf{x}_i$, since a locally flat function will lead to less output noise than a steep function, under the same location uncertainty.

[7] For the sake of notation, all integrals in this section are written as indefinite integrals, however they should be understood as definite integrals over appropriate sets.

Remark 1. This optimization involves high computational complexity and possibly numerical instability (due to the sum of exponentials). More importantly, a good estimate of $\boldsymbol{\theta}$ can only be found if a sample $\mathbf{X}^{(m)}$ is generated that is close to the true locations $\mathbf{X}$. Due to the high dimensionality [37, Section 29.2], this is unlikely, even for large M. Hence, (13) will lead to poor estimates of $\hat{\boldsymbol{\theta}}$.

2) Prediction: Given the training database $\{\mathbf{U}, \mathbf{y}\}$ and $\hat{\boldsymbol{\theta}}$, we wish to determine $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*)$ for an uncertain test location with associated distribution $p(\mathbf{x}_*)$, described by $\mathbf{u}_*$. The posterior predictive distribution $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*)$ is obtained by integrating $p(P_{\mathrm{RX}}(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{x}_*)$ with respect to $\mathbf{X}$ and $\mathbf{x}_*$:

$$p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*) = \int p(P_{\mathrm{RX}}(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{x}_*)\, p(\mathbf{X})\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{X}\, \mathrm{d}\mathbf{x}_*. \quad (14)$$

This integral is again analytically intractable. The Laplace approximation was utilized in [20] to solve (14), while here we again resort to a Monte Carlo method by drawing M i.i.d. samples $\mathbf{X}^{(m)} \sim p(\mathbf{X})$ and $\mathbf{x}_*^{(m)} \sim p(\mathbf{x}_*)$, so that

$$\begin{aligned} &p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*) \\ &\approx \frac{1}{M} \sum_{m=1}^{M} p(P_{\mathrm{RX}}(\mathbf{x}_*^{(m)})|\mathbf{X}^{(m)}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{x}_*^{(m)}) \\ &= \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(P_{\mathrm{RX}}(\mathbf{x}_*^{(m)}); \bar{P}_{\mathrm{RX}}(\mathbf{x}_*^{(m)}), V_{\mathrm{RX}}(\mathbf{x}_*^{(m)})). \end{aligned} \quad (15)$$

As M increases, the approximate distribution will tend to the true distribution. We refer to (13) and (15) as Monte Carlo GP (MCGP). From (15), we can compute the mean ($\bar{P}_{\mathrm{RX}}^{\mathrm{MC}}(\mathbf{u}_*)$) and the variance ($V_{\mathrm{RX}}^{\mathrm{MC}}(\mathbf{u}_*)$) [38, Eq. (14.10) and Eq. (14.11)] as

$$\bar{P}_{\mathrm{RX}}^{\mathrm{MC}}(\mathbf{u}_*) = \frac{1}{M} \sum_{m=1}^{M} \bar{P}_{\mathrm{RX}}(\mathbf{x}_*^{(m)}) \quad (16)$$

$$V_{\mathrm{RX}}^{\mathrm{MC}}(\mathbf{u}_*) = \frac{1}{M} \sum_{m=1}^{M} \Big(\bar{P}_{\mathrm{RX}}(\mathbf{x}_*^{(m)}) - \bar{P}_{\mathrm{RX}}^{\mathrm{MC}}(\mathbf{u}_*)\Big)^2 + \frac{1}{M} \sum_{m=1}^{M} V_{\mathrm{RX}}(\mathbf{x}_*^{(m)}). \quad (17)$$

Remark 2. Prediction is numerically straightforward, though it involves the inversion of an $N \times N$ matrix $\mathbf{K}$ for each of the M samples $\mathbf{X}^{(m)}$. In the case training locations are known, we can utilize cGP to obtain a good estimate of $\boldsymbol{\theta}$ and efficiently and accurately compute $\bar{P}_{\mathrm{RX}}^{\mathrm{MC}}(\mathbf{u}_*)$ and $V_{\mathrm{RX}}^{\mathrm{MC}}(\mathbf{u}_*)$. When both training and test locations are known, the above procedure reverts to cGP.

B. Gaussian Approximation

We have seen that while MCGP can account for location uncertainty during prediction, it will fail to deliver adequate estimates of $\boldsymbol{\theta}$ during learning (see Remark 1). To address this, we can modify $p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta})$ from (11) using a Gaussian approximation through moment matching. In addition, we can also form a Gaussian approximation of $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*)$ for prediction. We will term this approach Gaussian approximation GP (GAGP). The expressions that are obtained in the learning of GAGP, namely the expectation of mean and covariance functions will be used later in the design of uncertain GP (described in Section V.C).

1) Learning: Given the training database $\{\mathbf{U}, \mathbf{y}\}$, the mean of $p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta})$ is given by

$$\begin{aligned} \mathbb{E}[\mathbf{y}|\mathbf{U}, \boldsymbol{\theta}] &= \iint \mathbf{y}\, p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\, p(\mathbf{X})\, \mathrm{d}\mathbf{X}\, \mathrm{d}\mathbf{y} \\ &= \iint (\mathbf{y}\, p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\, \mathrm{d}\mathbf{y})\, p(\mathbf{X})\, \mathrm{d}\mathbf{X} \\ &= \int \boldsymbol{\mu}(\mathbf{X})\, p(\mathbf{X})\, \mathrm{d}\mathbf{X} \\ &= \boldsymbol{\mu}(\mathbf{U}), \end{aligned} \quad (18)$$

where $\boldsymbol{\mu}(\mathbf{U}) = [\mu(\mathbf{u}_1), \mu(\mathbf{u}_2), \ldots, \mu(\mathbf{u}_N)]^{\mathrm{T}}$ and $\mu(\mathbf{u}_i) = \int \mu(\mathbf{x}_i)\, p(\mathbf{x}_i)\, \mathrm{d}\mathbf{x}_i$. The covariance matrix of $p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta})$ can be expressed as

$$\begin{aligned} &\mathrm{Cov}[\mathbf{y}, \mathbf{y}|\mathbf{U}, \boldsymbol{\theta}] \\ &= \int \mathbf{y}\mathbf{y}^{\mathrm{T}}\, p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\, p(\mathbf{X})\, \mathrm{d}\mathbf{X}\, \mathrm{d}\mathbf{y} - \boldsymbol{\mu}(\mathbf{U})\boldsymbol{\mu}(\mathbf{U})^{\mathrm{T}} \\ &= \int \big(\mathbf{K} + \boldsymbol{\mu}(\mathbf{X})\boldsymbol{\mu}(\mathbf{X})^{\mathrm{T}}\big)\, p(\mathbf{X})\, \mathrm{d}\mathbf{X} - \boldsymbol{\mu}(\mathbf{U})\boldsymbol{\mu}(\mathbf{U})^{\mathrm{T}} \\ &= \mathbf{K}_{\mathrm{u}} + \Delta, \end{aligned} \quad (19)$$

where $[\mathbf{K}_{\mathrm{u}}]_{ij} = C_{\mathrm{u}}(\mathbf{u}_i, \mathbf{u}_j) + \sigma_n^2\, \delta_{ij}$ in which

$$C_{\mathrm{u}}(\mathbf{u}_i, \mathbf{u}_j) = \int C(\mathbf{x}_i, \mathbf{x}_j)\, p(\mathbf{x}_i)\, p(\mathbf{x}_j)\, \mathrm{d}\mathbf{x}_i\, \mathrm{d}\mathbf{x}_j \quad (20)$$

and Δ is a diagonal matrix with entries

$$[\Delta]_{ii} = \int \mu^2(\mathbf{x}_i) p(\mathbf{x}_i)\, \mathrm{d}\mathbf{x}_i - \mu^2(\mathbf{u}_i). \quad (21)$$

We will refer to $\mu(\mathbf{u}_i)$ and $C_{\mathrm{u}}(\mathbf{u}_i, \mathbf{u}_j)$ as the *expected mean* and *expected covariance* function. We can now express the likelihood function as $p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta}) \approx \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}(\mathbf{U}), \mathbf{K}_{\mathrm{u}} + \Delta)$, so that $\boldsymbol{\theta}$ can be estimated by minimizing the negative log-likelihood function

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \Big\{ -\log(\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}(\mathbf{U}), \mathbf{K}_{\mathrm{u}} + \Delta)) \Big\}. \quad (22)$$

Remark 3. Learning in GAGP involves computation of the expected mean in (18) and (21), as well as the expected covariance function in (20). These integrals are generally again intractable, but there are cases where closed-form expression exist [17], [18]. These will be discussed in detail in Section V.C. GAGP avoids the numerical problems present in MCGP and will hence generally be able to provide a good estimate of $\boldsymbol{\theta}$.

2) Prediction: Given the training database $\{\mathbf{U}, \mathbf{y}\}$ and $\hat{\boldsymbol{\theta}}$, we approximate the predictive distribution $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*)$ by a Gaussian with mean $\bar{P}_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*)$

and variance $V_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*)$. These are given by

$$\begin{aligned}
&\bar{P}_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*) \\
&= \mathbb{E}[P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*] \\
&= \int \bar{P}_{\mathrm{RX}}(\mathbf{x}_*)\, p(\mathbf{X})\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{X}\, \mathrm{d}\mathbf{x}_* \\
&= \mu(\mathbf{u}_*) + \sum_{i=1}^{N} \int \beta_i\, C(\mathbf{x}_*, \mathbf{x}_i)\, p(\mathbf{X})\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{X}\, \mathrm{d}\mathbf{x}_*. \quad (23)
\end{aligned}$$

Note that β_i is itself a function of all $\mathbf{X}$'s and $\mathbf{x}_*$. Similarly $V_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*)$ is calculated as

$$\begin{aligned}
&V_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*) \\
&= \mathbb{E}[P_{\mathrm{RX}}^2(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*] - \bar{P}_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*)^2 \quad (24) \\
&= \int \Big(V_{\mathrm{RX}}(\mathbf{x}_*) + \bar{P}_{\mathrm{RX}}(\mathbf{x}_*)^2 \Big)\, p(\mathbf{X})\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{X}\, \mathrm{d}\mathbf{x}_* \\
&\quad - \bar{P}_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*)^2. \quad (25)
\end{aligned}$$

Note that both $\bar{P}_{\mathrm{RX}}(\mathbf{x}_*)$ and $V_{\mathrm{RX}}(\mathbf{x}_*)$ are functions of $\mathbf{X}$ (see (9)–(10)).

Remark 4. Prediction in GAGP requires complex integrals to be solved in (23)–(25) for which no general closed-form expressions are known. Hence, a reasonable approach is to use GAGP to learn $\hat{\boldsymbol{\theta}}$ and MCGP for prediction.

Remark 5. In case training locations are known, i.e., $\mathbf{U} \in \mathcal{X}$, (23) reverts to

$$\bar{P}_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*) = \mu(\mathbf{u}_*) + \sum_{i=1}^{N} \beta_i \int C(\mathbf{x}_*, \mathbf{x}_i)\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{x}_* \quad (26)$$

and (25) becomes

$$\begin{aligned}
&V_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*) \\
&= k_{**} - \sum_{i,j=1}^{N} [\mathbf{K}^{-1}]_{ij} \int C(\mathbf{x}_*, \mathbf{x}_i)\, C(\mathbf{x}_*, \mathbf{x}_j)\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{x}_* \\
&+ \int \mu(\mathbf{x}_*)^2\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{x}_* + 2 \sum_{i=1}^{N} \beta_i \Big(\int \mu(\mathbf{x}_*)\, C(\mathbf{x}_*, \mathbf{x}_i) \\
&\times p(\mathbf{x}_*)\, \mathrm{d}\mathbf{x}_* \Big) + \sum_{i,j=1}^{N} \beta_i \beta_j \int C(\mathbf{x}_*, \mathbf{x}_i)\, C(\mathbf{x}_*, \mathbf{x}_j)\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{x}_* \\
&- \bar{P}_{\mathrm{RX}}^{\mathrm{GA}}(\mathbf{u}_*)^2, \quad (27)
\end{aligned}$$

both of which can be computed in closed form, under some conditions, when $\mu(\mathbf{x})$ is constant in $\mathbf{x}$ [18, Section 3.4]. When both $\mathbf{U} \in \mathcal{X}$ and $\mathbf{u}_* \in \mathcal{X}$, GAGP reverts to cGP.

C. Uncertain GP

While GAGP avoids the learning problems inherent to MCGP, prediction is generally intractable. Hence, GAGP is not a fully coherent approach to deal with location uncertainty. To address this, we consider a new type of GP (uGP), which *operates directly on the location distributions, rather than on the locations*. uGP involves a mean function $\mu_{\mathrm{uGP}}(\mathbf{u}_i) : \mathcal{U} \to \mathbb{R}$ and a positive semidefinite covariance function $C_{\mathrm{uGP}}(\mathbf{u}_i, \mathbf{u}_j) : \mathcal{U} \times \mathcal{U} \to \mathbb{R}^+$, which considers as inputs $\mathbf{u} \in \mathcal{U}$ and as outputs $y \in \mathbb{R}$. In other words,

$$P_{\mathrm{RX}}(\mathbf{u}_i) \sim \mathcal{GP}(\mu_{\mathrm{uGP}}(\mathbf{u}_i), C_{\mathrm{uGP}}(\mathbf{u}_i, \mathbf{u}_j)). \quad (28)$$

The mean function is given by $\mu_{\mathrm{uGP}}(\mathbf{u}_i) = \mathbb{E}_{\mathbf{x}_i}[\mathbb{E}_{\Psi(\mathbf{x}_i)}[P_{\mathrm{RX}}(\mathbf{x}_i)]]$, already introduced as the expected mean function in (18). However, for the mean function to be useful in a GP context, it should be available in closed form. As in cGP, we have significant freedom in our choice of covariance function. Apart from all technical conditions on the covariance function as described in [12], it is desirable to have a covariance function that (i) is available in closed form; (ii) leads to decreasing correlation with increasing input uncertainty (even when both inputs have same mean); (iii) can account for varying amounts of input uncertainty; (iv) reverts to a covariance function of the form (4) when $\mathbf{u} \in \mathcal{X}$, (v) does not depend on the mean function $\mu(\mathbf{x})$. We will now describe the mean function $\mu_{\mathrm{uGP}}(\mathbf{u}_i)$ and covariance function $C_{\mathrm{uGP}}(\mathbf{u}_i, \mathbf{u}_j)$ in detail.

The mean function: According to law of iterated expectations, the mean function $\mu(\mathbf{u}_i)$ is expressed as

$$\mu(\mathbf{u}_i) = L_0 - 10\, \eta\, \mathbb{E}_{\mathbf{x}_i}[\log_{10}(\|\mathbf{x}_i\|)]. \quad (29)$$

While there is no closed-form expression available for (29), we can form a polynomial approximation $\sum_{j=0}^{J} a_j \|\mathbf{x}_i\|^j \approx \log_{10}(\|\mathbf{x}_i\|)$, where the coefficients a_j are found by least squares minimization. For a given range of $\|\mathbf{x}_i\|$, this approximation can be made arbitrarily close by increasing the order J. When $p(\|\mathbf{x}_i\|)$ is approximately Gaussian (which may be the case for $\|\mathbf{x}_i\| \gg 0$), $\mu(\mathbf{u}_i) \approx L_0 - 10\, \eta \sum_{j=0}^{J} a_j\, \mathbb{E}_{\mathbf{x}_i}[\|\mathbf{x}_i\|^j]$ can be evaluated in closed form, since all Gaussian moments are known. See Appendix A for details on the approximation.

The covariance function: While any covariance function meeting the criteria (i)–(v) listed above can be chosen, a natural choice is (see Section IV.A)

$$\begin{aligned}
C_{\mathrm{uGP}}(\mathbf{u}_i, \mathbf{u}_j) &= \mathrm{Cov}[P_{\mathrm{RX}}(\mathbf{x}_i), P_{\mathrm{RX}}(\mathbf{x}_j)|\mathbf{u}_i, \mathbf{u}_j] \\
&= \mathrm{Cov}[y_i, y_j|\mathbf{U}, \boldsymbol{\theta}] - \delta_{ij}\sigma_n^2. \quad (30)
\end{aligned}$$

Unfortunately, as we can see from (19), this choice does not satisfy criterion (v). An alternative choice is the expected covariance function $C_{\mathrm{u}}(\mathbf{u}_i, \mathbf{u}_j)$ from (20). This choice clearly satisfies criteria (ii), (iii), (iv), and (v). To satisfy (i), we can select appropriate covariance functions, tailored to the distributions $p(\mathbf{x}_i)$, or appropriate distributions $p(\mathbf{x}_i)$ for a given covariance function. Examples include:

- Polynomial covariance functions for Gaussian $p(\mathbf{x}_i)$ [17], [18].
- Covariance functions of the form (4) with $p = 1$, $\mathbf{x}_i \in \mathbb{R}$, for Laplacian $p(\mathbf{x}_i)$.
- Covariance functions of the form (4) with $p = 2$, $\mathbf{x}_i \in \mathbb{R}^2$, for Gaussian $p(\mathbf{x}_i)$ (i.e., $p(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i; \mathbf{z}_i, \boldsymbol{\Sigma}_i)$). The expected covariance function is then given by [17], [18]

$$\begin{aligned}
&C_{\mathrm{uGP}}(\mathbf{u}_i, \mathbf{u}_j) = C_{\mathrm{u}}(\mathbf{u}_i, \mathbf{u}_j) = \delta_{ij}\, \sigma_{\mathrm{proc}}^2 \quad (31) \\
&+ \sigma_\Psi^2 \Big| \mathbf{I} + d_c^{-2}(\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j)(1 - \delta_{ij}) \Big|^{-1/2} \\
&\times \exp\left(-\frac{1}{d_c^2} (\mathbf{z}_i - \mathbf{z}_j)^{\mathrm{T}} (\mathbf{I} + d_c^{-2}(\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j))^{-1} (\mathbf{z}_i - \mathbf{z}_j) \right).
\end{aligned}$$

Note that the factor $|\mathbf{I} + d_c^{-2}(\mathbf{\Sigma}_i + \mathbf{\Sigma}_j)(1-\delta_{ij})|^{-1/2}$ ensures that inputs $i \neq j$ with the same mean (i.e., $\mathbf{z}_i = \mathbf{z}_j$) exhibit lower correlation with increasing uncertainty. The factor $(\mathbf{I} + d_c^{-2}(\mathbf{\Sigma}_i + \mathbf{\Sigma}_j))^{-1}$ ensures that the measurements taken at locations with low uncertainty (smaller than d_c) can be explained by a large value of d_c, while for measurements taken at locations with high uncertainty, $C_{\mathrm{u}}(\mathbf{u}_i, \mathbf{u}_j)$ will be small and decreasing with increasing uncertainty.

1) Learning: Given the training database $\{\mathbf{U}, \mathbf{y}\}$ and choosing $\mu_{\mathrm{uGP}}(\mathbf{u}_i) = \mu(\mathbf{u}_i)$ and $C_{\mathrm{uGP}}(\mathbf{u}_i, \mathbf{u}_j) = C_{\mathrm{u}}(\mathbf{u}_i, \mathbf{u}_j)$, the model parameters are found by minimizing the log-likelihood function

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \{-\log(p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta})\} \\ &= \arg\min_{\boldsymbol{\theta}} \{-\log(\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}(\mathbf{U}), \mathbf{K}_{\mathrm{u}})\}. \end{aligned} \tag{32}$$

Note that in contrast to GAGP, we have constructed uGP so that $\boldsymbol{\mu}(\mathbf{U})$ and $\mathbf{K}_{\mathrm{u}}$ are available in closed form, making numerical minimization tractable.

Remark 6. Learning of uGP (32) corresponds to the case of learning (22) in GAGP for $\Delta = \mathbf{0}$ (e.g., for constant mean processes).

2) Prediction: Let $\bar{P}_{\mathrm{RX}}(\mathbf{u}_*)$ be the mean and $V_{\mathrm{RX}}(\mathbf{u}_*)$ be the variance of the posterior predictive distribution $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*)$ of uGP with uncertain training and test locations, then $p(P_{\mathrm{RX}}(\mathbf{u}_*)|\mathbf{U}, \mathbf{y}, \hat{\boldsymbol{\theta}}, \mathbf{u}_*) = \mathcal{N}(P_{\mathrm{RX}}(\mathbf{u}_*); \bar{P}_{\mathrm{RX}}(\mathbf{u}_*), V_{\mathrm{RX}}(\mathbf{u}_*))$. The expressions for $\bar{P}_{\mathrm{RX}}(\mathbf{u}_*)$ and $V_{\mathrm{RX}}(\mathbf{u}_*)$ are now in standard GP form:

$$\bar{P}_{\mathrm{RX}}(\mathbf{u}_*) = \mu(\mathbf{u}_*) + \mathbf{k}_{\mathrm{u}*}^{\mathrm{T}} \mathbf{K}_{\mathrm{u}}^{-1} (\mathbf{y} - \boldsymbol{\mu}(\mathbf{U})) \tag{33}$$

$$V_{\mathrm{RX}}(\mathbf{u}_*) = k_{\mathrm{u}**} - \mathbf{k}_{\mathrm{u}*}^{\mathrm{T}} \mathbf{K}_{\mathrm{u}}^{-1} \mathbf{k}_{\mathrm{u}*}, \tag{34}$$

where $\mathbf{k}_{\mathrm{u}*}$ is the $N \times 1$ vector of cross-covariances $C_{\mathrm{u}}(\mathbf{u}_*, \mathbf{u}_i)$ between the received power at the test distribution $\mathbf{u}_*$ and at the training distribution $\mathbf{u}_i$, and $k_{\mathrm{u}**}$ is the a priori variance $C_{\mathrm{u}}(\mathbf{u}_*, \mathbf{u}_*)$.

Remark 7. In case the training locations are known, i.e., $\mathbf{U} \in \mathcal{X}$, the mean $\bar{P}_{\mathrm{RX}}(\mathbf{u}_*)$ and the variance $V_{\mathrm{RX}}(\mathbf{u}_*)$ can be obtained from the expressions (33) and (34), respectively, by setting $\mathbf{\Sigma}_i = \mathbf{0}, \forall\, i \in \{1, 2, \ldots, N\}$. Furthermore, the resulting mean $\bar{P}_{\mathrm{RX}}(\mathbf{u}_*)$ is exactly the same as (26), obtained in GAGP. However, due to a different choice of covariance function, the predicted variance $V_{\mathrm{RX}}(\mathbf{u}_*)$ is different from (27).

Remark 8. When the test location is known, i.e., $\mathbf{u}_* \in \mathcal{X}$, the mean $\bar{P}_{\mathrm{RX}}(\mathbf{x}_*)$ and the variance $V_{\mathrm{RX}}(\mathbf{x}_*)$ are obtained from (33) and (34) by setting $\mathbf{\Sigma}_* = \mathbf{0}$.

D. Unified View

We are now ready to recap the main differences between cGP and uGP, and to provide a unified view of the four methods (cGP, MCGP, GAGP, and uGP). Fig. 3 describes the main processes in uGP and cGP, along with the inputs and outputs during the learning and prediction processes. The four methods are depicted in Fig. 4: all four methods revert to cGP when training and predictions occur in $\mathcal{X}$, i.e., when there is no uncertainty about the locations. MCGP is able to consider general input distributions in $\mathcal{U}$, but leads to non-Gaussian output distributions. Through a Gaussian approximation of these output distributions, GAGP can consider general inputs and directly determine a Gaussian output distribution. Both of these approaches (MCGP and GAGP) have in common that they treat the process with input $\mathbf{x} \in \mathcal{A}$ as a GP. In contrast, uGP treats the process with input $\mathbf{u} \in \mathcal{U}$ as a GP. This allows for a direct mapping from inputs in $\mathcal{U}$ to Gaussian output distributions. In terms of tractability for learning and prediction, the four methods are compared in Table I. We see that among all four methods, uGP combines tractability with good performance.

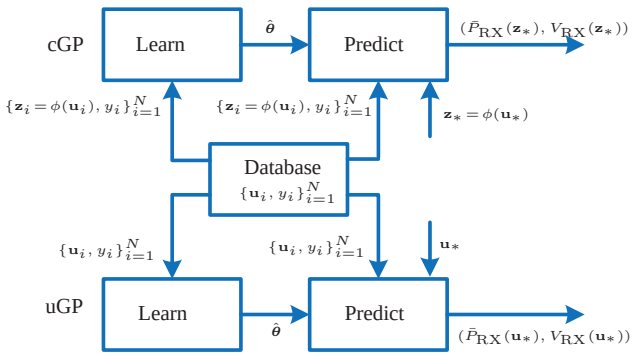

Figure 3. Learning and prediction phases of cGP and uGP. The difference in learning in uGP compared to cGP is that it considers location uncertainty of the nodes. The estimated model parameters $\hat{\boldsymbol{\theta}}$ are derived during the learning phase and are generally different in cGP compared to uGP. The mean $\bar{P}_{\mathrm{RX}}(\mathbf{z}_*)$ and variance $V_{\mathrm{RX}}(\mathbf{z}_*)$ of the posterior predictive distribution in cGP corresponds to a location $\mathbf{z}_*$ extracted from $\mathbf{u}_*$, which in turn represents $p(\mathbf{x}_*)$. In contrast, the mean $\bar{P}_{\mathrm{RX}}(\mathbf{u}_*)$ and variance $V_{\mathrm{RX}}(\mathbf{u}_*)$ of the posterior predictive distribution in uGP pertains to the entire location distribution represented by $\mathbf{u}_*$.

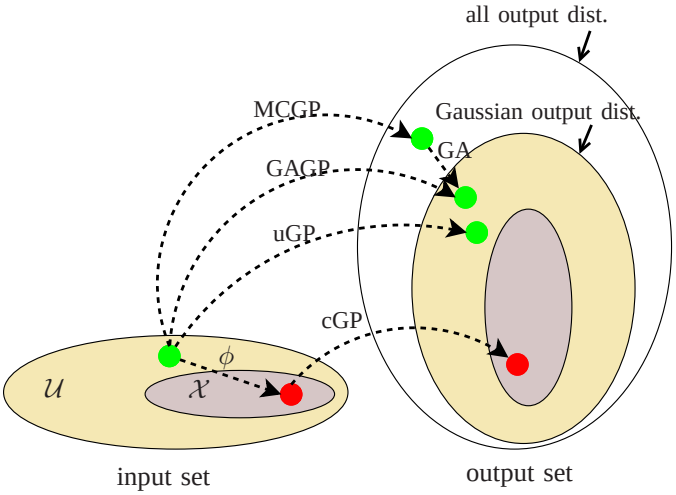

Figure 4. Relation between cGP, MCGP, GAGP, and uGP. All methods are equivalent when the input is limited to $\mathcal{X}$ (grey shaded area).

VI. Numerical Results and Discussion

In this section, we show learning and prediction results of cGP, uGP, and MCGP with uncertainty in training or test

Table I
COMPARISON OF TRACTABILITY FOR CGP, MCGP, GAGP, AND UGP IN LEARNING AND PREDICTION.

Method	Learning	Prediction
cGP	tractable, poor quality	closed-form, poor quality
MCGP	complex, poor quality	tractable
GAGP	tractable in some cases	intractable
uGP	tractable by design	closed-form

locations. In Section VI.D, we describe a resource allocation problem, where communication rates are predicted at future locations using cGP and uGP, in the presence of location uncertainty during training. The numerical analysis carried in this section is based on simulated channel measurements according to the model outlined in Section III.

Table II
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
η	2.5	M	300
σ_n	0.01	L_0	-10 dBm
d_c	15 m	σ_Ψ	10 dB

A. Simulation Setup

A geographical region $\mathcal{A}$ is considered and a base station is placed at the origin. A one dimensional radio propagation field is generated with sampling locations at a resolution of 0.25 m using an exponential covariance function $C_{\text{ref}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_\Psi^2 \exp\Big(-\|\mathbf{x}_i - \mathbf{x}_j\|/d_c\Big)$, corresponding to the Gudmundson model. Small-scale fading is assumed to have been averaged out[8]. The simulation parameters used to obtain the numerical results are given in Table II. We assume isotropic localization errors, so that $\mathbf{\Sigma}_i = \sigma_i^2\,\mathbf{I}$. To capture the effect of heterogeneous location errors, we draw the location error standard deviations from an exponential distribution, i.e., $\sigma_i \overset{\text{i.i.d.}}{\sim} \text{Exp}(\lambda)$, where λ is the average location error standard deviation. For cGP and MCGP, in order to not provide any unfair advantage to uGP, we use a covariance function of the form (4) with $p = 1$, in order to match the true covariance function $C_{\text{ref}}(\mathbf{x}_i, \mathbf{x}_j)$. For uGP, we use (31). Since uGP exhibits a mismatch in the covariance function, we absorb this mismatch in σ_{proc}, which is learned offline (more on this in Appendix B). We assume nodes know σ_n and L_0, which be inferred using standard methods [36], [39], [40], so they are not included in the learning process.

B. Learning Under Location Uncertainty

Fig. 5 depicts the impact of location uncertainty on the learning of hyperparameters $[d_c, \sigma_\Psi, \sigma_{\text{proc}}, \eta]$ for cGP, uGP, and MCGP. The learning of the hyperparameters is detailed in Appendix B.

[8]In the case small-scale fading is not averaged out, the proposed framework cannot be applied.

1) cGP: We first consider a variant of cGP, denoted as cGP-no-proc, in which σ_{proc} is fixed to zero. In cGP-no-proc, when $\lambda = 0$, the estimate $\hat{d}_c$ is non-zero. However, it can be observed in Fig. 5 (a), that with increase in λ, $\hat{d}_c$ decreases quickly to zero. Hence, cGP-no-proc will model the GP as a white process with high variance $\hat{\sigma}_\Psi^2$ and thus cannot handle the location uncertainty. On the other hand, in cGP where we estimate σ_{proc}, $\hat{\sigma}_{\text{proc}}$ absorbs part of location uncertainty (see Fig. 5 (c)). Consequently, the part of the observations that must be explained through σ_Ψ is reduced, leading to a reduction of $\hat{\sigma}_\Psi$ with λ. Due to this, cGP considers the measurements constitute a slowly varying process, therefore $\hat{d}_c$ increases with λ. An interesting observation is that the error bars for $\hat{d}_c$ also increase with λ. Hence, among cGP-no-proc and cGP, only cGP can reasonably deal with location uncertainty.

2) MCGP: The behavior is similar to that of cGP, i.e., an increase in $\hat{d}_c$, and a decrease in $\hat{\sigma}_\Psi$, when increasing λ. However, $\hat{\sigma}_\Psi$ decreases more quickly with λ when compared to cGP. These effects can be attributed to two causes: first of all, the inherent problem of drawing a finite number of samples as detailed at the end of Section V.A1; secondly, the fluctuations in the estimated path loss exponent $\hat{\eta}$ with increasing λ (see Fig. 5 (d)). The error bars of the estimates in this case are even higher than in cGP. As expected, MCGP is not suitable for learning.

3) uGP: As mentioned before, in uGP σ_{proc} is determined offline. The uGP model has the capability to absorb the location uncertainty into the covariance function. Due to this flexibility, it can handle higher values of λ and still maintain an almost constant $\hat{d}_c$ and $\hat{\sigma}_\Psi$ with increase in λ. For fair comparison with cGP, we also consider the case where σ_{proc} is estimated as part of the learning, referred to as uGP-proc. It can be observed in Fig. 5 (c) that $\hat{\sigma}_{\text{proc}}$ increases with increase in λ. When comparing uGP-proc to uGP, we observe a lower value of $\hat{\sigma}_\Psi$ and higher values of $\hat{d}_c$ and $\hat{\sigma}_{\text{proc}}$ for a particular value of λ. From this, we conclude that uGP should be preferred over uGP-proc, as it can explain the observations with smaller $\hat{\sigma}_{\text{proc}}$ and leads to simpler optimization. Finally, note that the error bars of the uGP estimates are relatively small when compared to cGP.

C. Prediction Under Location Uncertainty

Four cases can be considered, depending on whether training or testing inputs are in $\mathcal{X}$ or $\mathcal{U}$. We will focus on the case where *either* training or test locations are uncertain, but not both. From these, the behavior when both training and testing inputs are in $\mathcal{U}$ can be easily understood: only uGP can give reasonable performance among cGP, MCGP, and uGP, as the estimates of $\boldsymbol{\theta}$ in cGP and MCGP are of poor quality.

1) Uncertain training locations and certain testing locations: In this case $\mathbf{u}_i \in \mathcal{U}$ and $\mathbf{u}_* \in \mathcal{X}$. Fig. 6 (a) depicts the prediction results in terms of the predictive mean and predictive standard deviation (shown as shaded areas) for a particular realization of the channel field. It can be observed that uGP is able to predict the received power comparatively better than cGP and MCGP. uGP is able to estimate the underlying channel parameters better with the expected covariance

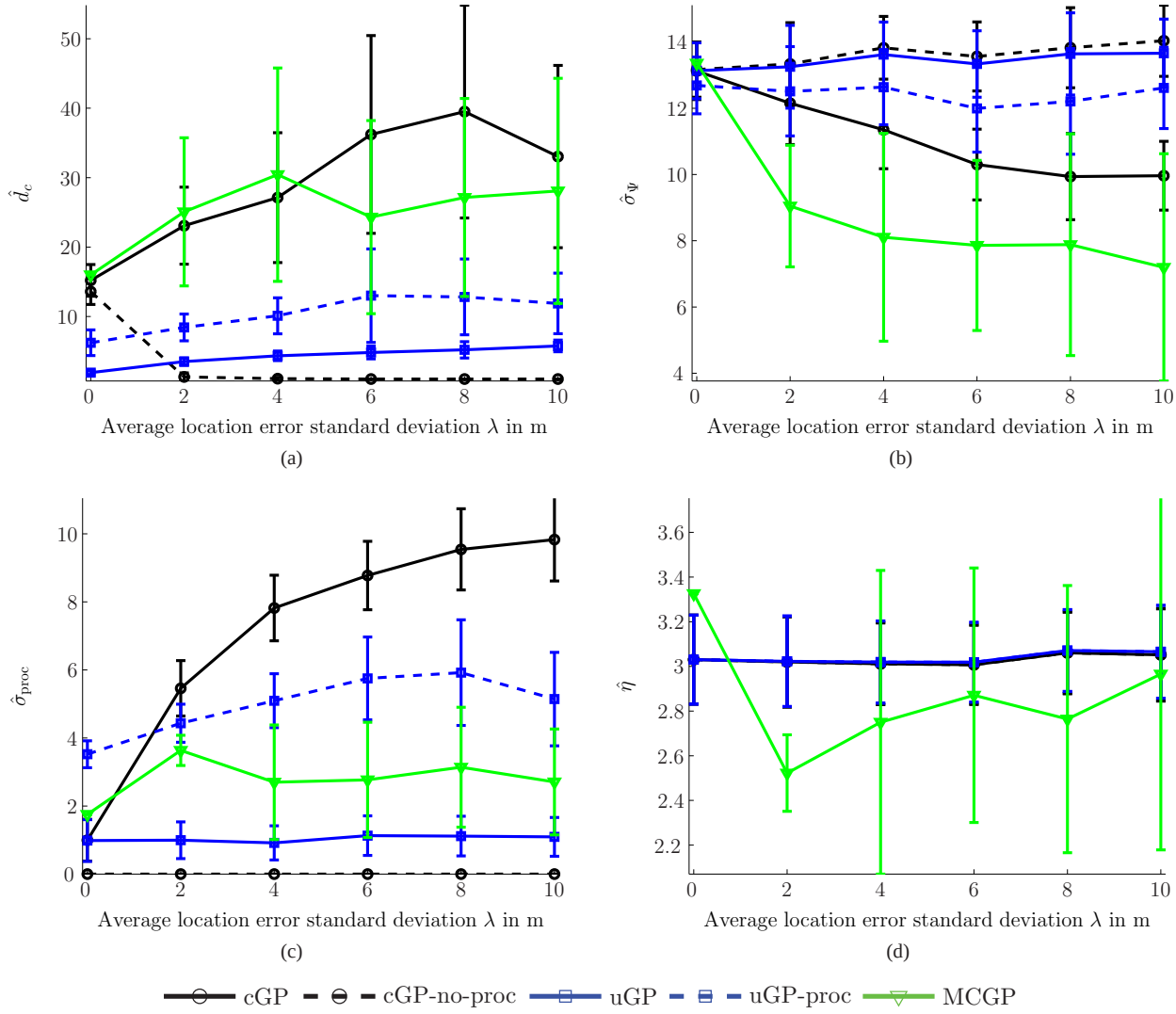

Figure 5. Impact of location uncertainty on learning the hyperparameters using cGP, uGP, and MCGP. The hyperparameters are estimated for each value of the mean location error standard deviation and for 40 realizations of the channel field. Results shown are the mean estimate of the hyperparameters and error bars with one standard deviation. Impact of location uncertainty in shown when estimating: (a) d_c, (b) σ_Ψ, (c) σ_{proc}, (d) η.

function, which takes in to account the location uncertainty of the nodes. In turn, this means that uGP can track the faster variations in the channel. cGP tries to model the true function with a slow varying process due to very high $\hat{d}_c$. Furthermore, cGP has higher uncertainty in predictions due to high $\hat{\sigma}_{\text{proc}}$ (see Fig. 5 (c)). On the other hand, MCGP has slightly better prediction performance (the standard deviation is not shown, but is slightly smaller than for cGP) compared to cGP due to the averaging by drawing samples from the distribution of the uncertain training locations. Averaging the prediction error over multiple channel realizations, Fig. 6 (b) shows the mean squared error (MSE) of the received power prediction of cGP and uGP with respect to λ (MCGP is not shown due to its similar performance to cGP). uGP clearly outperforms cGP (except fo $\lambda = 0$) due to its better tracking of the true channel (see Fig. 6 (a)) despite uncertainty on the training locations. The reason for higher MSE in the case of $\lambda = 0$ for uGP is due to its kernel mismatch.

2) Certain training locations and uncertain testing locations: In this case $\mathbf{u}_i \in \mathcal{X}$ and $\mathbf{u}_* \in \mathcal{U}$ (with a constant location error standard deviation σ m). Now the performance must be assessed with respect to the expected received power $P_{\text{RX,avg}}(\mathbf{u}_*) = \int P_{\text{RX}}(\mathbf{x}_*)\, p(\mathbf{x}_*)\, \mathrm{d}\mathbf{x}_*$, where $p(\mathbf{x}_*) = \mathcal{N}(\mathbf{z}_*, \sigma^2\, \mathbf{I})$, in which $\mathbf{z}_*$ is the mean of distribution described by $\mathbf{u}_*$. An example is shown in Fig. 7 (a), depicting $P_{\text{RX,avg}}$ as a function of $\mathbf{z}_*$, as well as the predictions from cGP, MCGP, and uGP. It can be observed that uGP and MCGP follow well $P_{\text{RX,avg}}$. Specifically, MCGP tracks $P_{\text{RX,avg}}$ quite closely as it is near-optimal in this case. In contrast, cGP follows the actual received power at $\mathbf{z}_*$, rather than the averaged power. This leads to fast variations in cGP, which are not present in uGP and MCGP. Fig. 7 (b) shows the MSE of the received power prediction of cGP, MCGP, and uGP with respect to σ when averaging the prediction error over multiple channel realizations. As expected, MCGP has the lower MSE than uGP and cGP. However, uGP performs better than cGP for all considered σ, except $\sigma = 0$ (due to kernel mismatch).

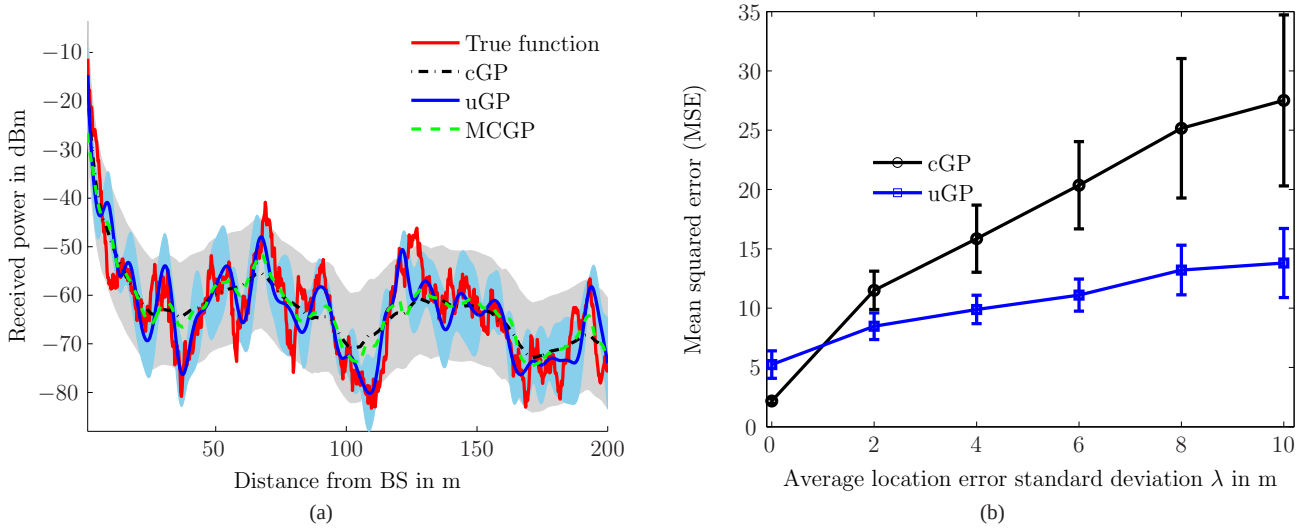

Figure 6. Performance comparison of cGP, MCGP, and uGP under uncertain training and certain testing locations. Inset (a) received power prediction using uncertain training locations with average location error of λ = 8 m and certain test locations for single realization of a channel field. The shaded area (grey for cGP and blue for uGP) depicts point wise predictive mean plus and minus the predictive standard deviation, and (b) MSE performance of cGP and uGP as a function of average location error standard deviation λ. The MSE is averaged for each value of λ and for 50 realizations of the channel field is shown are the mean of the MSE and error bars with one standard deviation. The MSE is calculated as $\frac{1}{|\mathcal{T}|}\sum_{\mathbf{x}_*\in\mathcal{T}}(P_{\mathrm{RX}}(\mathbf{x}_*)-\bar{P}_{\mathrm{RX}}(\mathbf{x}_*))^2$, where $\mathcal{T}$ is the set of test locations and $|\mathcal{T}|$ denotes its cardinality.

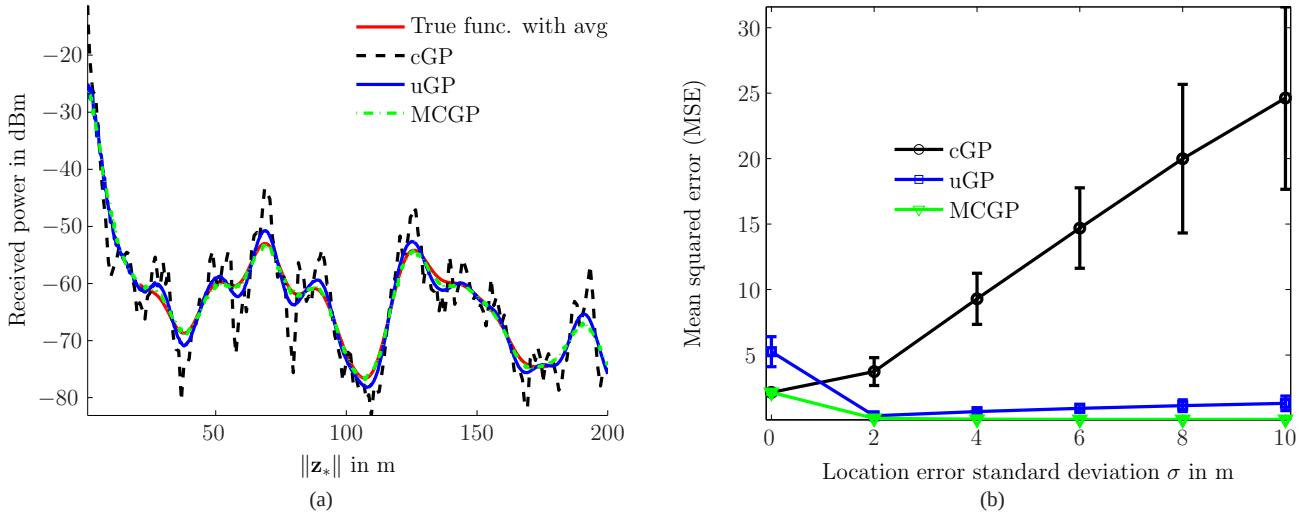

Figure 7. Performance comparison of cGP, MCGP, and uGP under certain training and uncertain testing locations. Inset (a) received power prediction using certain training and uncertain test locations with a constant location error standard deviation $\sigma = 5$ m for single realization of channel field, and (b) MSE performance of cGP, MCGP and uGP as a function of constant location error standard deviation σ on test locations. The MSE is averaged for each value of σ and for 50 realizations of the channel field is shown are the mean of the MSE and error bars with one standard deviation. The MSE is calculated as $\frac{1}{|\mathcal{T}^u|}\sum_{\mathbf{u}_*\in\mathcal{T}^u}(P_{\mathrm{RX,avg}}(\mathbf{u}_*)-\bar{P}_{\mathrm{RX}}(\mathbf{u}_*))^2$, where $\mathcal{T}^u$ is the set of test location distributions and $|\mathcal{T}^u|$ denotes its cardinality.

Furthermore, the performance of uGP is very close to that of MCGP.

D. Resource Allocation Example

1) Scenario: In this section, we compare cGP and uGP for a simple proactive resource allocation scenario. We consider a user moving through a region $\mathcal{A}$ and predict the CQM at each location. The supported rate, expressed in bits per channel use (bpu), for a user at location $\mathbf{x}_*$ is defined as

$$r(\mathbf{x}_*) = \log_2\big(1+\mathrm{SNR}(\mathbf{x}_*)\big), \tag{35}$$

where $\mathrm{SNR}(\mathbf{x}_*) = P^{\mathrm{lin}}_{\mathrm{RX}}(\mathbf{x}_*)/W^{\mathrm{lin}}$, is the signal-to-noise ratio at location $\mathbf{x}_*$, W^{lin} is the receiver thermal noise and $P^{\mathrm{lin}}_{\mathrm{RX}}(\mathbf{x}_*)$ is the received power, both measured in linear scale. The average rate in the region $\mathcal{A}$, denoted as $\bar{r}^{\mathrm{ref}}_{\mathcal{A}}$, is defined as

$$\bar{r}^{\mathrm{ref}}_{\mathcal{A}} = \frac{1}{|\mathcal{A}|}\int_{\mathcal{A}} r(\mathbf{x}_*)\mathrm{d}\mathbf{x}_*, \tag{36}$$

where $|\mathcal{A}|$ denotes area of the region $\mathcal{A}$. The predicted rate for a user at a future location $\mathbf{x}_*$, based on the predicted CQM values $(\bar{P}_{\mathrm{RX}}(\mathbf{x}_*), V_{\mathrm{RX}}(\mathbf{x}_*))$, is defined as

$$r(\mathbf{x}_*,\alpha) = \log_2\big(1+\mathrm{SNR}(\mathbf{x}_*,\alpha)\big), \tag{37}$$

where $\alpha \geq 0$ is a confidence parameter, $\mathrm{SNR}(\mathbf{x}_*,\alpha) = P^{\mathrm{lin}}_{\mathrm{RX}}(\mathbf{x}_*,\alpha)/W^{\mathrm{lin}}$ and $P_{\mathrm{RX}}(\mathbf{x}_*,\alpha) = 10\log_{10}\big(P^{\mathrm{lin}}_{\mathrm{RX}}(\mathbf{x}_*,\alpha)\big) = \bar{P}_{\mathrm{RX}}(\mathbf{x}_*) - \alpha\left(V_{\mathrm{RX}}(\mathbf{x}_*)\right)^{\frac{1}{2}}$.

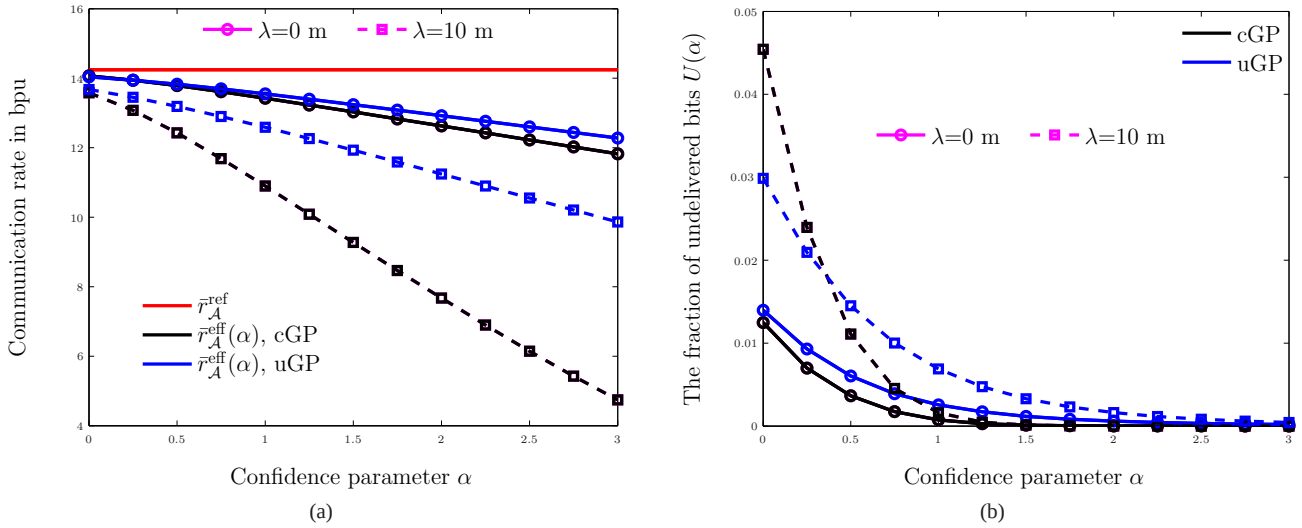

Figure 8. Resource allocation example for cGP, and uGP with two different values of localization error standard deviations ($\lambda \in \{0, 10\}$ m) and for different values of the confidence parameter α. The results are averaged for each value of λ with 50 channel realizations. Inset (a) the effective rate $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$, and (b) the fraction of undelivered bits $U(\alpha)$.

2) Performance measure: The user moves through the environment according to a known trajectory. The base station allocates bits to each future location, proportional to $r(\mathbf{x}_*, \alpha)$. When the user is at location $\mathbf{x}_*$, only a fraction of the bits, proportional to $\min(r(\mathbf{x}_*, \alpha), r(\mathbf{x}_*))$ would be delivered. Therefore, the effective rate $r^{\text{eff}}(\mathbf{x}_*, \alpha)$ for the user at location $\mathbf{x}_*$ is

$$r^{\text{eff}}(\mathbf{x}_*, \alpha) = \min(r(\mathbf{x}_*, \alpha), r(\mathbf{x}_*)). \tag{38}$$

The average effective rate $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$ for a given confidence level α is then computed by spatial average of $r^{\text{eff}}(\mathbf{x}_*, \alpha)$ over region $\mathcal{A}$ as

$$\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha) = \frac{1}{|\mathcal{A}|} \int_{\mathcal{A}} r^{\text{eff}}(\mathbf{x}_*, \alpha)\, \mathrm{d}\mathbf{x}_* \in [0, \bar{r}^{\text{ref}}_{\mathcal{A}}]. \tag{39}$$

When $r(\mathbf{x}_*, \alpha) > r(\mathbf{x}_*)$, a part of the allocated bits cannot be delivered. The total fraction of undelivered bits over the environment is given by

$$U(\alpha) = \frac{\int_{\mathcal{A}} (r(\mathbf{x}_*, \alpha) - r^{\text{eff}}(\mathbf{x}_*, \alpha))\, \mathrm{d}\mathbf{x}_*}{\int_{\mathcal{A}} r(\mathbf{x}_*, \alpha)\, \mathrm{d}\mathbf{x}_*} \in [0, 1). \tag{40}$$

Hence, $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$ describes the rate that the user will receive (penalizing under-estimation of the rate), while $U(\alpha)$ describes the loss due to lost bits (penalizing over-estimating of the rate).

3) Predicted communication rates with uncertain training locations: We predict the CQM at known test locations $\mathbf{x}_* \in \mathcal{X}$, based on training with uncertain locations (considering $\lambda \in \{0, 10\}$ m), all within a one-dimensional region $\mathcal{A}$. The average effective rate $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$ and the fraction of undelivered bits $U(\alpha)$, as a function of α, are shown in Fig 8 (a)–(b), respectively. As expected, increasing α leads to a more conservative allocation, thus reducing both $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$ and $U(\alpha)$. For a specific value of α, increase in λ decreases $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$. This is due to the fact that with increase in λ, the mean $\bar{P}_{\text{RX}}(\mathbf{x}_*)$ is of poor quality and the variance $V_{\text{RX}}(\mathbf{x}_*)$ is high for CQM predictions.

It is evident that when $\lambda = 0$, uGP and cGP attain similar performance, both in terms of $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$ and $U(\alpha)$. When λ is increased to 10 m, cGP suffers from a significant reduction in effective rate $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$, while at the same time dropping up to 4.5 % of the bits. This is due to cGP's poor predictions, which are either too low (leading to a reduction in $\bar{r}^{\text{eff}}_{\mathcal{A}}(\alpha)$) or too high (leading to an increase in $U(\alpha)$). In contrast, uGP, which is able to track the channel well despite uncertain training, achieves a higher effective rate, especially for high confidence values (e.g., around 2 times higher rate for $\alpha = 3$, for $U(\alpha)$ less than 0.1%).

VII. Conclusion

Channel quality metrics can be predicted using spatial regression tools such as Gaussian processes (GP). We have studied the impact of location uncertainties on GP and have demonstrated that, when heterogeneous location uncertainties are present, the classical GP framework is unable to (i) learn the underlying channel parameters properly; (ii) predict the expected channel quality metric. By introducing a GP that operates directly on the location distribution, we find uncertain GP (uGP), which is able to both learn and predict in the presence of location uncertainties. This translates in better performance when using uGP for predictive resource allocation.

Possible avenues of future research include validation using real measurements, modeling correlation of shadowing in the temporal dimension, study of better approximations for learning with uncertain locations, and the extension to ad-hoc networks.

Appendix A
Approximation of Expected Mean Function

Let $d_i = \|\mathbf{x}_i\|$ and recall from random variable transformation theory that

$$\int \log_{10}(\|\mathbf{x}_i\|)\, p(\mathbf{x}_i)\, \mathrm{d}\mathbf{x}_i = \int \log_{10}(d_i)\, p(d_i)\, \mathrm{d}d_i. \tag{41}$$

We assume $p(\mathbf{x}_i) = \mathcal{N}(\mathbf{z}_i, \sigma_i^2 \mathbf{I})$, so $p(d_i)$ follows a Rician distribution

$$p(d_i) = \frac{d_i}{\sigma_i^2} \exp\Big(-\frac{\|\mathbf{z}_i\|^2 + d_i^2}{2\,\sigma_i^2}\Big)\, I_0\Big(\frac{\|\mathbf{z}_i\|\, d_i}{\sigma_i^2}\Big)\; d_i > 0, \quad (42)$$

where $I_0(.)$ is a modified Bessel function of zero-th order. For $\|\mathbf{z}_i\|/\sigma_i \geq 3$, $p(d_i)$ can be approximated as a Gaussian distribution

$$p_{\text{Gauss}}(d_i) = \frac{1}{\sqrt{2\,\pi\sigma_i^2}} \exp\Big(-\frac{(\|\mathbf{z}_i\| - d_i)^2}{2\,\sigma_i^2}\Big). \quad (43)$$

The integral (41) still does not have a closed form expression with $p_{\text{Gauss}}(d_i)$. Now approximating the $\log_{10}(.)$ function with a polynomial function of the form $w(d_i) = \sum_{j=0}^{J} a_j\, d_i^j$ then (41) can be written as

$$\int \log_{10}(\|\mathbf{x}_i\|)\, p(\mathbf{x}_i)\, \mathrm{d}\mathbf{x}_i \approx \int_{-\infty}^{+\infty} w(d_i)\, p_{\text{Gauss}}(d_i)\, \mathrm{d}d_i, \quad (44)$$

which can be computed exactly.

Appendix B
Learning Procedure

In this appendix, we detail the learning of $\boldsymbol{\theta} = [\sigma_n, \sigma_{\text{proc}}, d_c, L_0, \eta, \sigma_\Psi]$ for cGP, uGP, and MCGP. We consider nodes know σ_n and L_0, therefore they are not estimated as part of the learning process. Let the remaining set of hyperparameters be $\boldsymbol{\theta} = [\sigma_{\text{proc}}, d_c, \sigma_\Psi]$ and η .

cGP

Based on Section III, we can write the received measurements $\mathbf{y}$ with their corresponding training locations $\mathbf{X}$ in matrix form as

$$\mathbf{y} = \mathbf{1}^{\mathrm{T}} L_0 + \mathbf{h}_{\mathrm{c}}\, \eta + \boldsymbol{\Psi} + \mathbf{n}, \quad (45)$$

where $\boldsymbol{\Psi} = [\Psi(\mathbf{x}_1), \dots \Psi(\mathbf{x}_N)]^{\mathrm{T}}$, $\mathbf{n} = [n_1, \dots, n_N]^{\mathrm{T}}$, and $\mathbf{h}_{\mathrm{c}} = -10\,[\log_{10}(\|\mathbf{x}_1\|), \dots, \log_{10}(\|\mathbf{x}_N\|)]^{\mathrm{T}}$. Assuming the measurements are uncorrelated, then the least squares estimate of the path-loss exponent can be computed as

$$\hat{\eta} = \big(\mathbf{h}_{\mathrm{c}}^{\mathrm{T}}\, \mathbf{h}_{\mathrm{c}}\big)^{-1}\, \mathbf{h}_{\mathrm{c}}^{\mathrm{T}}\, \big(\mathbf{y} - \mathbf{1}^{\mathrm{T}} L_0\big). \quad (46)$$

Once the path-loss exponent is estimated, the mean component of the received measurements can be subtracted as, $\boldsymbol{\Upsilon}_{\mathrm{c}} = \mathbf{y} - \mathbf{1}^{\mathrm{T}} L_0 - \mathbf{h}_{\mathrm{c}}\, \hat{\eta}$. Then, $\boldsymbol{\Upsilon}_{\mathrm{c}}$ becomes a zero-mean Gaussian process. Now the likelihood function (6) becomes $l(\boldsymbol{\theta}) = p(\boldsymbol{\Upsilon}_{\mathrm{c}}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\Upsilon}_{\mathrm{c}}; \mathbf{0}, \mathbf{K})$. The hyperparameters $\boldsymbol{\theta}$ are estimated by minimizing negative logarithm of $l(\boldsymbol{\theta})$

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \{-\log(p(\boldsymbol{\Upsilon}_{\mathrm{c}}|\mathbf{X}, \boldsymbol{\theta})\} \\ &= \arg\min_{\boldsymbol{\theta}} \Big\{\log|\mathbf{K}| + \boldsymbol{\Upsilon}_{\mathrm{c}}^{\mathrm{T}}\, \mathbf{K}^{-1}\, \boldsymbol{\Upsilon}_{\mathrm{c}}\Big\}. \end{aligned} \quad (47)$$

We calculate the variance of the process $\boldsymbol{\Upsilon}_{\mathrm{c}}$ as $\sigma_{\text{Tot}}^2 = 1/N \sum_{i=1}^{N} [\boldsymbol{\Upsilon}_{\mathrm{c}}]_i^2$. The variance of the process should be captured by the hyperparameters σ_{proc}, σ_n, and σ_Ψ. We define $\sigma_{\text{proc}}^2 = \sigma_{\text{Tot}}^2 - \sigma_n^2 - \sigma_\Psi^2$, as a result $l(\boldsymbol{\theta})$ becomes a function of only d_c and σ_Ψ. We solve (47) and find $\hat{d}_c$ and $\hat{\sigma}_\Psi$ by an exhaustive grid search. Once $\hat{d}_c$ and $\hat{\sigma}_\Psi$ are found, then $\hat{\sigma}_{\text{proc}}$ can be calculated as $\hat{\sigma}_{\text{proc}}^2 = \hat{\sigma}_{\text{Tot}}^2 - \sigma_n^2 - \hat{\sigma}_\Psi^2$.

uGP

In this case, the path-loss exponent is estimated as

$$\hat{\eta} = \big(\mathbf{h}_{\mathrm{u}}^{\mathrm{T}}\, \mathbf{h}_{\mathrm{u}}\big)^{-1}\, \mathbf{h}_{\mathrm{u}}^{\mathrm{T}}\, \big(\mathbf{y} - \mathbf{1}^{\mathrm{T}} L_0\big), \quad (48)$$

where $\mathbf{h}_{\mathrm{u}} = -10\,[\mathbb{E}_{\mathbf{x}_1}[\log_{10}(\|\mathbf{x}_1\|)], \dots, \mathbb{E}_{\mathbf{x}_N}[\log_{10}(\|\mathbf{x}_N\|)]^{\mathrm{T}}$. Once again removing the mean from the measurements, we obtain $\boldsymbol{\Upsilon}_{\mathrm{u}} = \mathbf{y} - \mathbf{1}^{\mathrm{T}} L_0 - \mathbf{h}_{\mathrm{u}}\, \hat{\eta}$. The hyperparameters $\boldsymbol{\theta}$ are estimated by minimizing the modified negative log-likelihood function

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \{-\log(p(\boldsymbol{\Upsilon}_{\mathrm{u}}|\mathbf{U}, \boldsymbol{\theta})\} \\ &= \arg\min_{\boldsymbol{\theta}} \Big\{\log|\mathbf{K}_{\mathrm{u}}| + \boldsymbol{\Upsilon}_{\mathrm{u}}^{\mathrm{T}}\, \mathbf{K}_{\mathrm{u}}{}^{-1}\, \boldsymbol{\Upsilon}_{\mathrm{u}}\Big\}. \end{aligned} \quad (49)$$

Again, $\sigma_{\text{Tot}}^2 = 1/N \sum_{i=1}^{N} [\boldsymbol{\Upsilon}_{\mathrm{u}}]_i^2$, is the variance of the process. As a result, $\hat{\sigma}_\Psi$ becomes $\hat{\sigma}_\Psi^2 = \sigma_{\text{Tot}}^2 - \sigma_n^2 - \hat{\sigma}_{\text{proc}}^2$ and due to this $l(\boldsymbol{\theta})$ is now only a function of d_c. We solve (49) and find $\hat{d}_c$ by an exhaustive grid search.

The learning process can be simplified for uGP: since σ_{proc} only captures kernel mismatch irrespective of the location uncertainty and path loss, the value of $\hat{\sigma}_{\text{proc}}$ can be obtained offline with noise-free training locations by performing learning as in the case of cGP, but with a covariance function of the form (4) for $p = 2$. This approach gives an advantage to cGP and thus makes the comparison between uGP and cGP more fair for all values of $\lambda \geq 0$.

MCGP

It is no longer feasible to estimate η first and subtract to make the process zero mean, because of summation in the Monte Carlo integration (12). Therefore, we optimize (13) with respect to the hyperparameters η and $\boldsymbol{\theta}$ using `fminsearch` function of Matlab.

Acknowledgment

The authors would like to thank Ido Nevat, Lennart Svensson, Ilaria Malanchini, and Vinay Suryaprakash for their feedback on the manuscript.

References

[1] S. Sand, R. Tanbourgi, C. Mensing, and R. Raulefs, "Position aware adaptive communication systems," in *Forty-Third Asilomar Conference on Signals, Systems and Computers*, 2009, pp. 73–77.

[2] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5G networks," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 102–112, Nov 2014.

[3] H. Abou-zeid, H. Hassanein, and S. Valentin, "Optimal predictive resource allocation: Exploiting mobility patterns and radio maps," in *IEEE Globecom Workshops*, Dec 2013, pp. 4877–4882.

[4] J. Johansson, W. Hapsari, S. Kelley, and G. Bodog, "Minimization of drive tests in 3GPP release 11," *IEEE Communications Magazine*, vol. 50, no. 11, pp. 36–43, 2012.

[5] A. Zalonis, N. Dimitriou, A. Polydoros, J. Nasreddine, and P. Mahonen, "Femtocell downlink power control based on radio environment maps," in *IEEE Wireless Communications and Networking Conference*, April 2012, pp. 1224–1228.

[6] A. Galindo-Serrano, B. Sayrac, S. Ben Jemaa, J. Riihijarvi, and P. Mahonen, "Harvesting MDT data: Radio environment maps for coverage analysis in cellular networks," in *International Conference on Cognitive Radio Oriented Wireless Networks*, July 2013, pp. 37–42.

[7] I. Nevat, G. Peters, and I. Collings, "Location-aware cooperative spectrum sensing via Gaussian processes," in *Australian Communications Theory Workshop*, Jan 2012, pp. 19–24.

[8] J. Tadrous, A. Eryilmaz, and H. El Gamal, "Proactive resource allocation: Harnessing the diversity and multicast gains," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 4833–4854, Aug 2013.

[9] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[10] N. Jalden, P. Zetterberg, B. Ottersten, A. Hong, and R. Thoma, "Correlation properties of large scale fading based on indoor measurements," in *IEEE Wireless Communications and Networking Conference*, March 2007, pp. 1894–1899.

[11] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2001.

[12] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[13] L. Csató and M. Opper, "Sparse on-line Gaussian processes," *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.

[14] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[15] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Advances in neural information processing systems*, vol. 18, p. 1257, 2006.

[16] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, 2013.

[17] P. Dallaire, C. Besse, and B. Chaib-draa, "An approximate inference with Gaussian process to latent functions from uncertain data," *Neurocomputing*, vol. 74, no. 11, pp. 1945–1955, 2011.

[18] A. Girard, "Approximate methods for propagation of uncertainty with Gaussian process models," Ph.D. dissertation, University of Glasgow, 2004.

[19] A. Girard and R. Murray-Smith, "Learning a Gaussian process model with uncertain inputs," Department of Computing Science, University of Glasgow, Tech. Rep. TR-2003-144, June 2003.

[20] M. Jadaliha, Y. Xu, J. Choi, N. Johnson, and W. Li, "Gaussian process regression for sensor networks under localization uncertainty," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 223–237, 2013.

[21] A. McHutchon and C. E. Rasmussen, "Gaussian process training with input noise," in *Advances in Neural Information Processing Systems*, 2011, pp. 1341–1349.

[22] A. Ranganathan, M.-H. Yang, and J. Ho, "Online sparse Gaussian process regression and its applications," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 391–404, Feb 2011.

[23] S.-J. Kim, E. Dall'Anese, and G. Giannakis, "Cooperative spectrum sensing for cognitive radios using kriged Kalman filtering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 24–36, 2011.

[24] D. Gu and H. Hu, "Spatial Gaussian process regression with mobile sensor networks," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1279–1290, Aug 2012.

[25] M. P. Deisenroth and J. W. Ng, "Distributed Gaussian processes," *arXiv preprint arXiv:1502.02843*, 2015.

[26] S. Choi, M. Jadaliha, J. Choi, and S. Oh, "Distributed Gaussian process regression for mobile sensor networks under localization uncertainty," in *52nd Annual Conference on Decision and Control*, Dec 2013, pp. 4766–4771.

[27] J. Quinonero Candela, "Learning with uncertainty-Gaussian processes and relevance vector machines," Ph.D. dissertation, Technical University of Denmark, 2004.

[28] J. Fink, "Communication for teams of networked robots," Ph.D. dissertation, Elect. Syst. Eng., Univ. Pennsylvania, Philadelphia, PA, Aug 2011.

[29] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 8, pp. 4024–4036, 2009.

[30] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 964–978, 2012.

[31] Y. Yan and Y. Mostofi, "Impact of localization errors on wireless channel prediction in mobile robotic networks," in *IEEE Globecom, Workshop on Wireless Networking for Unmanned Autonomous Vehicles*, Dec. 2013.

[32] G. L. Stüber, *Principles of Mobile Communication (2nd Ed.)*. Kluwer Academic Publishers, 2001.

[33] A. Goldsmith, L. Greenstein, and G. Foschini, "Error statistics of real-time power measurements in cellular channels with multipath and shadowing," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 439–446, Aug 1994.

[34] S. S. Szyszkowicz, H. Yanikomeroglu, and J. S. Thompson, "On the feasibility of wireless shadowing correlation models," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4222–4236, 2010.

[35] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics letters*, vol. 27, no. 23, pp. 2145–2146, 1991.

[36] Y. Mostofi, M. Malmirchegini, and A. Ghaffarkhah, "Estimation of communication signal strength in robotic networks," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 1946–1951.

[37] D. J. MacKay, *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[38] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[39] K. V. Mardia and R. Marshall, "Maximum likelihood estimation of models for residual covariance in spatial regression," *Biometrika*, vol. 71, no. 1, pp. 135–146, 1984.

[40] P. K. Kitanidis, "Statistical estimation of polynomial generalized covariance functions and hydrologic applications," *Water Resources Research*, vol. 19, no. 4, pp. 909–921, 1983.